(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 June 2001 (28.06.2001)

PCT

(10) International Publication Number
WO 01/46765 A1

(51) International Patent Classification[7]: G05B 19/418, 19/042

(21) International Application Number: PCT/US00/35258

(22) International Filing Date:
22 December 2000 (22.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/171,439 22 December 1999 (22.12.1999) US
09/664,154 18 September 2000 (18.09.2000) US

(71) Applicant: ROCKWELL TECHNOLOGIES, LLC [US/US]; 1049 Camino Dos Rios, Thousand Oaks, CA 91360 (US).

(72) Inventors: LENNER, Joseph, A.; 7578 Bendleton Drive, Hudson, OH 44056 (US). VASKO, David, A.; 9480 Woodview Drive, Macedonia, OH 44056 (US). VANDESTEEG, Kerry, W.; 75 West Bellmeadow, Chagrin Falls, OH 44022 (US). HALL, Kenwood, H.; 1768 East Sapphire Drive, Hudson, OH 44236 (US).

(74) Agent: BAXTER, Keith, M.; Quarles & Brady, LLP, 411 East Wisconsin Avenue, Milwaukee, WI 53202-4497 (US).

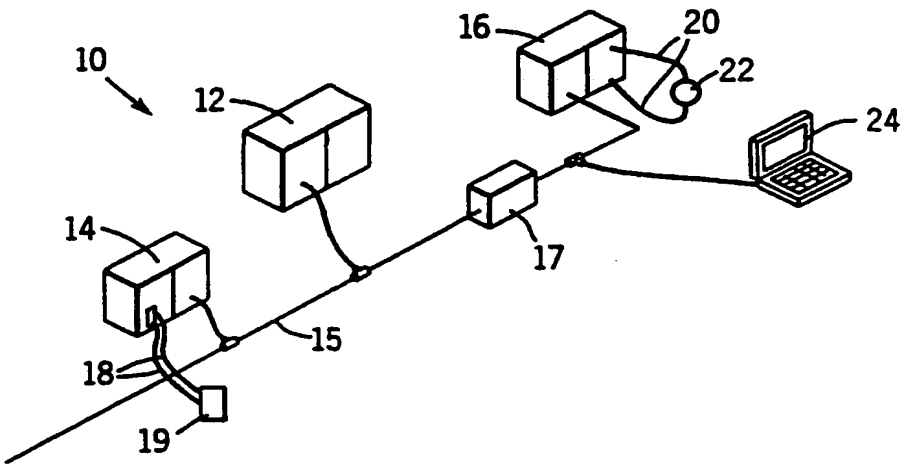
(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

[Continued on next page]

(54) Title: SAFETY NETWORK FOR INDUSTRIAL CONTROLLER HAVING REDUCED BANDWIDTH REQUIREMENTS



(57) Abstract: A highly reliable industrial control system is produced using a network running a standard serial protocol. A safety protocol is embedded within the standard serial protocol by adding to special error detecting data redundant with the protocol of the standard serial network. In addition an overarching protocol involving timing of messages is imposed to provide the necessary level of reliability in the standard serial network. The system may also use a standard network for configuration by symmetrically transmitting configuration data to intercommunicating controller components. The configuration data provides a unique identification to each component to reduce the possibility of misdirected messages and provides the protocols to be used by the components reducing the chance of garbled messages. Two connections may be used to transmit redundant data and include a reply message to the message producer. Comparison of the data of the two messages can reveal other types of failure not apparent by these former techniques. The system provides for the processing of redundant control signals on as little as a single serial network without overloading the network by preprocessing input signals for coincidence and sending only the coincidence signal either periodically or on a change of state.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SAFETY NETWORK FOR INDUSTRIAL CONTROLLER HAVING REDUCED BANDWIDTH REQUIREMENTS

This application claims the benefit of provisional application 60/171,439 filed on December 22, 1999.

CROSS-REFERENCE TO RELATED APPLICATIONS

--

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

--

BACKGROUND OF THE INVENTION

The present invention relates to industrial controllers used for real-time control of industrial processes, and in particular to high-reliability industrial controllers appropriate for use in devices intended to protect human life and health. "High reliability" refers generally to systems that guard against the propagation of erroneous data or signals by detecting error or fault conditions and signaling their occurrence and/or entering into a predetermined fault state. High reliability systems may be distinguished from high availability system, however, the present invention may be useful in both such systems and therefore, as used herein, high reliability should not be considered to exclude high availability systems.

Industrial controllers are special purpose computers used in controlling industrial processes. Under the direction of a stored control program, an industrial controller examines a series of inputs reflecting the status of the controlled process and changes a series of outputs controlling the industrial process. The inputs and outputs may be binary, that is, on or off, or analog, providing a value within a continuous range. The inputs may be obtained from sensors attached to the controlled equipment and the outputs may be signals to actuators on the controlled equipment.

"Safety systems" are systems intended to ensure the safety of humans working in the environment of an industrial process. Such systems may include the electronics associated with emergency stop buttons, interlock switches and machine lockouts. Traditionally, safety systems have been implemented by a set of circuits wholly separate from the industrial control system used to control the industrial process with which the

safety system is associated. Such safety systems are "hard-wired" from switches and relays, some of which may be specialized "safety relays" allowing comparison of redundant signals and providing internal checking of conditions such as welded or stuck contacts. Safety systems may use switches with dual contacts providing an early indication of contact failure, and multiple contacts may be wired to actuators so that the actuators are energized only if multiple contacts close.

Hard-wired safety systems have proven inadequate, as the complexity of industrial processes has increased. This is in part because of the cost of installing and wiring relays and in part because of the difficulty of troubleshooting and maintaining the "program" implemented by the safety system in which the logic can only be changed by rewiring physical relays and switches.

For this reason, there is considerable interest in implementing safety systems using industrial controllers. Such controllers are easier to program and have reduced installation costs because of their use of a high-speed serial communication network eliminating long runs of point-to-point wiring.

Unfortunately, high-speed serial communication networks commonly used in industrial control are not sufficiently reliable for safety systems. For this reason, efforts have been undertaken to develop a "safety network" being a high-speed serial communication network providing greater certainty in the transmission of data. Currently proposed safety networks are incompatible with the protocols widely used in industrial control. Accordingly, if these new safety networks are adopted, existing industrial controller hardware and standard technologies may be unusable, imposing high costs on existing and new factories. Such costs may detrimentally postpone wide scale adoption of advanced safety technology.

What is needed is a safety network that is compatible with conventional industrial controller networks and components. Ideally such a safety network would work with a wide variety of different standard communication protocols and would allow the mixing of standard industrial control components and safety system components without compromising reliability.

An additional problem in implementing safety systems over standard networks is that the redundant control signals used to detect failures in hard-wired systems (when they don't match) do not always change at exactly the same time. Accordingly a window of time is established during which lack of coincidence of the signals is ignored. Ideally this window is short so that actual failures can be quickly identified.

A short coincidence window creates problems, however, when a high reliability system is implemented on a standard serial network such as is used in control systems. This is because for reasonable network bandwidths, queuing of messages introduces skew in the transmission of the redundant signals, requiring an undesirable lengthening of the transmission window. This is particularly true when the communications of signals requires reply messages with separate network transmissions.

What is also needed is a safety network that is compatible with conventional industrial controller serial networks and components yet that provides the benefits that come from using redundant control signals. Ideally such a safety network would work the currently available bandwidths of industrial control networks.

BRIEF SUMMARY OF THE INVENTION

The present invention provides high reliability communications over standard control networks by opening redundant "connections" under the connected messaging protocols of such standard networks and by adopting an echoing of messages sent that reveals to both message producers and message consumers failure of either connection. Dual connections thus serve in lieu of dual media traditionally used in such systems making the imposition of high reliability possible with existing network media.

Specifically, the present invention provides a method of establishing high reliability communication among components of an industrial controller some of which receive control signals from a controlled process, the components communicating over a standard network. The method includes the steps of establishing at least two redundant logical message producers associated with a given received control signal and opening a logical connection between each of the two logical message producers and two corresponding logical message consumers. Data, including a given received control signal, is transmitted on the connections from the logical message producers to the logical message consumers and after receipt of uncorrupted data at each logical message consumer, transmitting reply data including the given received control signal on the connection to the logical message producers. The logical message producers respond to an absence of an uncorrupted receipt of a transmission of reply data by entering a predetermined safety state.

Thus it is one object of the invention to provide for high reliability communications under standard connected messaging communications protocols. The

redundant connections and reply messages provide resistance to undetected message corruption.

The present invention further provides a high-reliability communications system that can make use of standard networks for initialization.

One requirement of a high reliability system is that messages not be mis-directed. This ordinarily can be assure by giving each communicating device a way of identifying itself and making sure that each device establishes the identity of all other parties with whom it communicates. Ideally, the identities will be unique to a given "connection" or communication pair of one message producer and one message consumer.

Another requirement is that all parties know the parameters of communication. Errors in communication parameters can cause messages to be misinterpreted or unintelligible.

The need to notify each device of its identity and to communicate common communication parameters is best met by transmitting parameters and identities to the devices over the standard network as the high reliability communications system is initialized. Unfortunately, the distribution of identities and parameters over a standard network can work against establishing a high reliability communications system, if there is appreciable chance that the identities or parameters will be mis-directed or garbled.

The present invention allows the configuration of a highly reliable communications system over a standard network by use of a configuration tool (possibly a separate device) symmetrically communicating configuration data to two devices intended to communicate with each other during control time. The configuration data provides both identities to the communicating parties (unique to a connection or communication pair) and also coveys important parameters of the communication. After receiving the configuration data, the two intercommunicating parties may compare configuration data to ensure that they are correctly part of a connection.

Specifically, the present invention provides a method of establishing high reliability communication among components of an industrial control system exchanging control signals with a controlled process, the components communicating over a standard network. The method includes the first step of transmitting a configuration message from a configuration source to a first component and a second component over the standard network using a standard network protocol, the configuration message providing data related to a high reliability communications protocol usable on the standard network. In a next step, the configuration source receives a configuration response message from the

first component and the second component, the configuration response message describing data of a configuration message previously received by the first component and the second component. Communication of control signals between the first and second component, as defined by the data of the configuration message, is enabled only if the configuration response message received by the configuration source describes the same data as the configuration message transmitted from the configuration source.

Thus it is one object of the invention to permit a standard network to be used to configure and identify devices that will be communicating as part of a high reliability communications system. The symmetrical transmission of the configuration data to the two intercommunicating devices and the need for a response message reflecting the configurations data reduces many types of errors to which standard networks are prone.

The present invention further provides a network-independent, high-reliability communications system by imposing two levels of protocol on data being transmitted over the network. The first level of protocol provides the high-reliability necessary for a safety system while the second level of protocol encapsulates the protocol of the safety system allowing it to be transmitted on a standard network.

The safety network of the present invention is compatible with a wide variety of standard networks and standard network hardware and also allows safety and non-safety components to be intermixed through the medium of the standard network. The safety protocol adds error detection data to transmitted data, all of which is treated as data by the standard network. The safety protocol also adds timing requirements for message transmission that ensure reliable transmission but that can remain invisible to the protocol of the standard network.

Specifically the present invention provides a network-independent, high-reliability communication system for an industrial controller making use of a standard serial communication network. The communication system includes a first I/O communication circuit receiving I/O data for control of an industrial process. This I/O data may be input or output data and the circuit may be at an industrial controller or at its I/O module. The I/O data is received by a first network-independent protocol circuit which formats it for transmission under a network-independent protocol to produce high-reliability formatted data, formatted so as to reduce errors in transmission. The high-reliability formatted data is received by a standard network protocol circuit, which further formats it for transmission under a protocol of the standard serial communication network. In this way, it produces doubly-formatted data for transmission on the standard serial communication

network. The formatting of the protocol for the standard serial communication network also reduces errors in transmission. After transmission over a standard serial communication network, the doubly-formatted data is received by a second standard network protocol circuit, which extracts the high-reliability formatted data according to the protocol of the standard serial communication network. The data is then received by a second network-independent protocol circuit extracting from the high-reliability formatted data, the original I/O data. A second I/O communication circuit receives the I/O data for control of the industrial process.

Thus it is one object of the invention to provide for a high-reliability communication system that may nevertheless use the standard circuitry and protocols of common, well-developed and commercially available serial communication networks. The safety formatting is embedded in the standard protocol of the serial network, which treats the formatted safety data as normal data to be transmitted.

Finally, the present invention facilitates the transmission and use of redundant control signals on standard serial networks by moving the coincidence detection step to the message producers prior to transmission of the control signal on the network. A single coincidence signal is developed with a short coincidence window that may then be redundantly transmitted over the network. Because the coincidence is resolved prior to transmission, network skew does not require a lengthening of the coincidence window.

Specifically, the present invention provides a high reliability industrial control system having a controller with a first network interface to a shared serial network. The industrial control system also includes an input module with at least two interface circuits for receiving at least two redundant input signals, the interface circuits communicating with at least one processor via an internal bus. The processor further communicating with a second network interface to the shared serial network and executes a stored program to: receive the redundant input signals processed by the interface circuits; determine a coincidence of the redundant input signals within a window of a predefined time period; and only when there is coincidence within the window, transmit via the second network interface, at least one coincidence signal indicating a coincident state of the redundant input signals to the controller.

Thus it is one object of the invention to permit the use of a relatively short predefined time period for the coincidence window by eliminating the effect of network skew of the input signals.

The foregoing and other objects and advantages of the invention will appear from the following description. In the description, reference is made to the accompanying drawings, which form a part hereof, and in which there is shown by way of illustration a preferred embodiment of the invention. Such embodiment does not necessarily represent the full scope of the invention, however, and reference must be made to the claims herein for interpreting the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a perspective view of a simplified industrial controller using a standard serial communication network linking a central controller with remote input and output circuits and with a remote configuration terminal, such as may be used in the present invention;

Fig. 2 is a schematic block diagram of the control system of Fig. 1 showing redundant wiring from an input switch to the input circuit of Fig. 1, the input circuits having redundant components such as may process the signals from the input switch to send signals over the communication network to the controller of Fig. 1, the controller having redundant processors to send signals over the communications network to the output circuit of Fig. 1, the output circuit having redundant components to provide outputs to an actuator;

Fig. 3 is a fragmentary view similar to Fig. 2, showing an alternative configuration of the input circuit of Fig. 2 using conventional control input circuits without redundant components;

Fig. 4 is a fragmentary view similar to Fig. 2, showing an alternative configuration of the output circuit of Fig. 2 using conventional control output circuits without redundant components;

Fig. 5 is a representational view of the dual communication protocols provided by the present invention in which data is first encoded with a safety protocol and then with a network protocol to be compatible with the serial network;

Fig. 6 is a schematic representation of a data word transmitted over the standard serial network showing the embedding of safety formatting data with I/O data within the formatting provided by the standard serial network;

Fig. 7 is a graphical representation having time on the vertical axis and distance along the network on the horizontal axis, showing transmission of configuration messages

to the input circuit, the controller and the output circuit, forming the foundation of the safety protocol of the present invention;

Fig. 8 is a figure similar to that of Fig. 7 showing the transmission of messages after the configuration process during a start-up and run-time phase of the network;

Fig. 9 is a block diagram of the industrial controller of Fig. 1 showing the division of communications between the input circuit, the controller and the output circuit into producer-consumer pairs such as provides redundant communication over a single network and the varied topologies of the implementations of Figs. 2, 3 and 4;

Fig. 10 is a flow chart showing the principle stages of the safety protocol of initialization, start-up, and run-time;

Fig. 11 is a figure similar to that of Fig. 7 showing normal protocol operation under the safety protocol of the present invention during run-time;

Fig. 12 is a figure similar to Fig. 11 showing protocol operation with a corrupted producer message;

Fig. 13 is a figure similar to Fig. 11 showing protocol operation with a lost producer message;

Fig. 14 is a figure similar to Fig. 11 showing protocol operation with a corrupted acknowledgement message from the consumer;

Fig. 15 is a figure similar to Fig. 11 showing protocol operation with a lost consumer acknowledgement message;

Fig. 16 is a figure similar to Fig. 11 showing protocol operation with disruption of the connection between the producer and consumer;

Fig. 17 is a graph of a typical input signal over time showing a skew resulting from different sampling points of two redundant input circuits;

Fig. 18 is a program that may be executed by the input circuit of Fig. 1 for eliminating the skew of Fig. 17 prior to network transmission;

Fig. 19 is a flow chart of a program executed by the producers of Fig. 9 in implementing the safety protocol; and

Fig. 20 is a flow chart of a program executed by the consumers of Fig. 9 in implementing the safety protocol of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention can be part of a "safety system" used to protect human life and limb in the industrial environment. Nevertheless, the term "safety" as used herein is

not a representation that the present invention will make an industrial process safe or that other systems will produce unsafe operation. Safety in an industrial process depends on a wide variety of factors outside of the scope of the present invention including: design of the safety system, installation and maintenance of the components of the safety system, and the cooperation and training of individuals using the safety system. Although the present invention is intended to be highly reliable, all physical systems are susceptible to failure and provision must be made for such failure.

Referring now to Fig. 1, an industrial control system 10 for implementing a safety system with the present invention includes a controller 12 communicating on a serial network 15 with remote input module 14 and remote output module 16. The network 15 may be a standard and commonly available high-speed serial network including but not limited to: Ethernet, DeviceNet, ControlNet, Firewire or FieldBus. The network 15 may optionally include a bridge 17 translating between different of the above standard or other protocols. As will be understood from the following, the present invention may be easily adapted to bridge applications.

Input module 14 may accept input signals 18 (on like-designated lines) which are communicated over the network 15 to the industrial controller 12. At the industrial controller 12 the signals 18 may be processed under a control program implementing a safety system (such as a machine lock-out or emergency stop) and further signals sent to the output module 16 which may produce output signals 20 (on like-designated lines) to an actuator 22.

The input signals 18 may come from a switch 19 which may be any of a variety of devices producing safety input signals including but not limited to emergency stop switches, interlock switches, light curtains and other proximity detectors. The actuator 22 may be a relay, solenoid, motor, enunciator, lamp or other device implementing a safety function.

Also connected to the network 15 is a standard computer, which may be used as a configuration terminal 24 whose purposes will be described below.

Redundant System Hardware

Referring now to Fig. 2, the switch 19 may produce redundant signals 18a and 18b where signal 18a is, for example, from a first contact within the switch 19, and signal 18b is from a second independent contact within switch 19. The contacts may have the same logic (as shown) both being normally open (e.g., closed with actuation of a pushbutton 26) or may be inverted logic with one contact normally open and one contact normally

closed. In either case, redundant signals 18a and 18b are generated so as to provide for higher reliability in the determining the state of the switch 19.

The input module 14 may include redundant interface circuitry 28a receiving signal 18a and interface circuitry 28b receiving signal 18b. Alternatively, but not shown, interface circuitry 28a and 28b may each receive both signal 18a and 18b (for internal comparison) or may receive signals 18a and 18b from a single contact. The contacts, in generating signals 18a and 18b, may each be provided with a separate voltage from the input circuitry 28a and 28b or from a common voltage source (not shown). Other redundant variations on these wiring systems, known in the art, may also be used.

Each of the interface circuitry 28a and 28b may in turn provide signals to associated microcontrollers 30a and 30b. Microcontrollers 30a and 30b provide a computer processor, memory and a stored program for executing safety protocol programs as will be described below. Alternatively, or in addition, the safety protocol may be executed by safety protocol circuits 32 with which microcontrollers 30a and 30b communicate. In this case, the safety protocol circuits 28a and 28b may be application-specific integrated circuits (ASIC). As it is well known in the art to implement protocols through hardware or software or combinations of each, the term "protocol device" as used herein and in the claims should be understood to embrace generally any combination of software and hardware components implementing the indicated functions.

The microcontrollers 30a and 30b may communicate with each other through an internal bus 34 to compare signals 18a and 18b as will be described.

Microcontrollers 30a and 30b or safety protocol circuits 28a and 28b in turn connect to standard network protocol circuits 36a and 36b of a type well known in the art for handling the low level protocol of the standard network 15. Typically, the standard network protocol circuits 36a and 36b are implemented by an ASIC whose implementation represents considerable development time and which cannot be easily modified.

The standard network protocol circuits 36a and 36b transmits signals from the input module 14 on the network 15 to be received at the controller 12 through a similar standard network protocol circuits 38a and 38b. These signals are processed by the standard network protocol circuit 38 and provided to redundant safety protocol circuits 40a and 40b, being similar to safety protocol circuits 32a and 32b described before. These safety protocol circuits 40a and 40b communicate with processors 42a and 42b, respectively, which include separate memory systems and control programs according to

well-known redundancy techniques and which intercommunicate on internal bus 34'. Output signals generated by the processors 42a and 42b may be communicated back through the safety protocol circuits 40a and 40b to implement the safety protocol, as will be described below (or alternatively, the safety protocol may be handled by the processor 42a and 42b), and the output signals communicated to the standard network protocol circuits 38a and 38b for transmission again on network 15 to output module 16.

Output module 16 may receive output data through a standard network protocol circuits 44a and 44b being similar to standard network protocol circuits 36a and 36b and 38a and 38b. The standard network protocol circuits 44a and 44b provide the data to safety protocol circuits 46a and 46b, which in turn provide them to redundant controllers 48a and 48b. As before, alternatively, the safety protocol may be handled by the controllers 48a and 48b instead. The controllers 48a and 48b communicate by internal bus 34" and in turn provide signals to output interface circuits 50a and 50b which provide the output signals 20a and 20b. The output signals may be connected to the actuator 22 so that outputs must be enabled for the actuator 22 to be powered. In this sense, a default safety state is produced (of no power to the actuator 22) if there is an inconsistency between the signals received by processors 48a and 48b. A change in the wiring to parallel configurations could create a safety state where the actuator is actuated unless both signals received by processors 48a and 48b are not enabled.

Alternatively, and as will be described, a safety state may be enforced by a safety state signal transmitted from the controller 12 or the input module 14 to the microcontrollers 48a and 48b of output module 16, the latter which may respond by producing outputs to output interface circuits 50a and 50b determined by stored values of desired safety states programmed through the configuration terminal 24 as will be described further below.

A bridge circuit 17 per the present invention could use the basic structure shown in the input module 14 but replacing the interface circuitry 28a and 28b of input module 14 with network protocol circuits 38a and 38b and safety protocol circuits of 40a and 40b (where the network protocol circuits 38 and 36 are for different protocols, thereby allowing seamless transmission of safety data per the techniques described below).

Referring now to Fig. 3, specialized redundant input module 14, in the present invention, may be replaced with two standard input modules 14a and 14b, input module 14a holding the equivalent of previously described interface circuitry 28a, microcontroller 30a, safety protocol circuit 32a and standard network protocol circuit 36a; and input

module 14b holding the equivalent of interface circuitry 28b, microcontroller 30b, safety protocol circuit 32b, and standard network protocol circuit 36b. In this case, the operation of safety protocol circuits 32a and 32b are implemented in the firmware of the microcontrollers 30a and 30b and effected via messages communicated on the network 15 rather than the internal bus 34.

Likewise, referring to Fig. 4, the redundancy of output module 16 may be implemented by separate output circuits 16a and 16b, output module 16a including the equivalent of standard network protocol circuit 44, safety protocol circuit 46a, microcontroller 48a, and output interface circuit 50a, with output module 16b including the equivalents of standard network protocol circuit 44 as 44', safety protocol circuit 46b, microcontroller 48b, and output interface circuit 50b. As will be described below, the present invention provides a protocol that is indifferent to the exact parsing of the safety components among physical devices having addresses on the network 15.

Referring now to Figs. 5 and 2, the operation of the safety protocol circuits 32 and standard network protocol circuits 36 in the input circuit is to embed input data 52 from lines 18b within a safety-network protocol 54 implemented both as additional data attached to messages sent on network 15 and in the management of that data as will be described. The safety-network protocol 54 is in turn encapsulated in the standard network protocol 56 for seamless transmission on the network 15.

The Safety Network Protocol

Referring now to Figs. 5 and 2, the operation of the safety protocol circuits 32, 40 and 46 in conjunction with the standard network protocol circuits 36, 38 and 44 is to embed I/O data 52 (e.g., from lines 18b) within a safety-network protocol 54 implemented both as additional data attached to I/O data 52 sent on network 15 and in the management of the particulars of transmission of that I/O data 52. The safety-network protocol 54 is in turn encapsulated in the standard network protocol 56 for seamless transmission on the network 15.

The data encapsulated in the safety-network protocol 54 and standard network protocol 56 can then be received (e.g., by the controller 12) and extracted through the successive operation of the standard network protocol circuits 36, 38 and 44 and the safety protocol circuits 32, 40 and 46 to provide the I/O data 52 in its basic state. Note that Fig. 5 is only symbolic of the process and that the safety-network protocol 54 is not simply an encapsulation of the data 52 within for example safety data headers but rather the safety protocol includes timing constraints that may be executed in sequence with the

standard network protocol 56 so that the safety-network protocol 54 may operate within the standard network protocol 56 without modification of the network 15 or standard network protocol circuits 36, 38 and 44.

This dual level encapsulation and de-encapsulation is performed for each transmission of I/O data 52 on the network 15 that requires a high level of reliability commensurate with safety systems. For non-safety system data, the standard network protocol 56 may be used alone without the safety-network protocol 54 for communication with non-safety elements of the industrial control system 10. Because all data transmitted on the network 15 is embedded in the standard network protocol 56, the safety-network protocol 54 will work seamlessly with a variety of networks 15 providing they have data transmission capacity suitable for the I/O data 52 and sufficient in capacity to accept some added safety error detection data 58 of the safety-network protocol 54 as will be described.

Safety Message Formatting

Referring now to Fig. 6, a first aspect of the safety-network protocol 54 is that the I/O data 52 is attached to safety error detection data 58 to form a safety message 60 that forms the data provided to the standard network protocol circuits 36, 38 and 44 to produce a network message 61. The safety error detection data 58 may include a sequence count indicating the local order in which the safety message 60 is transmitted with respect to earlier transmissions of safety messages. The sequence count is normally limited in range (0-3) as it is intended, as will be described, to detect the loss of only a single message.

Also appended to the I/O data 52 and part of the safety error detection data 58 is a cyclic redundancy code (CRC) selected in the preferred embodiment to be twelve-bits. The cyclic redundancy code is functionally generated from the I/O data 52 and the sequence count so that an error in the transmission of either of those data elements can be detected when the CRC is recalculated by the receiving device and doesn't match. As is understood in the art, a twelve bit error code will allow the detection of errors with odd numbers of bit errors, all two-bit errors, all burst errors up to twelve bits and 99.951% of burst errors larger than twelve bits, for up to two-thousand and forty seven bits of data of the safety message 60.

The safety message 60 is embedded in the network headers and footers 62 and 64, which vary depending on the standard network protocol 56 of the network 15. Depending on the network 15, the network header and footer 62 and 64 may include a CRC code and

sequence count and other similar safety error detection data 58 operating redundantly with the safety error detection data 58. Nevertheless, the safety message 60 includes its own safety error detection data 58 so as to be wholly network-independent to the degree possible.

Connected Messaging

As mentioned above, the safety error detection data 58 forms only part of the safety-network protocol 54. The safety-network protocol 54 also includes a configuration step that ensures proper communication under a connected messaging scheme. Referring now to Fig. 9, the communications between the controller 12, input module 14 (or input modules 14a and 14b) and the output module 16 (or output module 16a and 16b) may provide a connected messaging system. As is understood in the art, connected messaging involves opening a connection between pairs of logical devices one that acts as a "producers" of a message and one that acts as a "consumers" of the message. The process of opening the connection reserves bandwidth of the network and reserves necessary processing and buffering resources at the producer and consumer to ensure that data of the connection will be reliably transmitted and received.

The connected messaging protocol may be implemented as part of the safety network protocol 54 or as part of the standard network protocol 56, the latter option limiting somewhat the types of standard networks 15 that may be used. Some standard network protocols that support connected messaging are DeviceNet and Control Net, Ethernet, and ATM.

Referring now to Fig. 9, under a connected messaging protocol, the input module 14 provides two producers 80 opening two connections with two consumers 82 of the controller 12, one for each of the signals 18a and 18b. As a practical matter, these two connections mean that two separate network messages 61 will be sent over the network 15 thus decreasing the chance of loss of both messages.

For the implementation of Fig. 3 with separate input module 14a and 14b, two producers 80 are also provided. Even though the producers 80 are now in different devices (having different addresses on the network 15), the operation of the control program implementing the safety system, above the connection level, need not changed by these changes in implementations. Connected messaging thus makes the safety system largely indifferent to topologies as providing for a natural redundancy over a single network, or multiple links

Controller 12 likewise includes two producers 80 exchanging data with consumers 82 either in a single output module 16 per Fig. 2 or in separate output module 16a and 16b per the embodiment of Fig. 4. Two arrows are shown between each producer 80 and consumer 82 indicating the paring of each message with an acknowledgment message under the safety protocol 54 as will be described below, per Fig. 9.

The bridge circuit 17, not shown in Fig. 9, but as described above, would implement four consumers and four producers (two for each network side) as will be understood to those of ordinary skill in the art.

Safety Configuration Data and Protocol

Referring now to Fig. 10, the safety protocol more generally includes an initialization state, of which the first step is developing configuration data as indicated by process block 66.

The configuration process involves developing configuration data at the configuration terminal 24 and ensuring that accurate copies of that configuration data are at each of the input module 14, the controller 12, and the output module 16. The configuration data is unique to each connection, provides essential components of the safety protocol, and identifies intercommunicating parties so as to reduce the possibility of improper connections injecting spurious data into the safety system. This is particularly important in allowing mixing of systems components observing the safety network protocol 54 with standard components observing only the standard network protocol. Devices may support multiple connections, in which case multiple configuration data specific to each connection will be used.

Generally, the configuration data include data uniquely identifying the particular device of the input module 14, the controller 12, and the output module 16 holding the configuration data, and particularly the serial number of that device. The serial number is a unique and immutable part of the physical devices and thus together with an internal address of the logical devices within the physical device (which may establish independent connections) the serial number provides each connection with a unique identity eliminating the possibility of crossed connections between different devices once the configuration data is properly disseminated. To augment the serial number, the configuration data may also include a vendor identification number, a device code, a product code, major revision, minor revision, as well as network data including the logical, physical address of the device, all known in the art and identifying the particular

device. Similarly, the configuration data within a device may include the serial number of the device to which it is connected.

As mentioned, the connection data may also include data necessary for the implementation of the other aspects of the safety protocol as are yet to be described, including variables of "periodic time interval", "reply timer interval", "filter count", and "retry limit". The configuration data also includes the safety state to which the device will revert in the case of network error and a list of related I/O points indicating other I/O points (related to other connections), which should revert to the safety state if the present connection has an error. This later feature allows selective and intelligent disabling of the safety system upon a communication error as will be described. As will be evident from context, some of this data is dependent on the devices and the system programmer must develop some.

Referring to Fig. 7, configuration data held within the configuration terminal 24 is sent to each of the input module 14, the controller 12, and the output module 16 as messages 66a, 66b and 66c.

The receiving input module 14, the controller 12, and the output module 16 store the configuration and respond with the same configuration message but changed to a one's complement form (being simply a different binary coding (the inversion)) of the configuration data received. This one's complement message is returned by messages 66d, 66e, and 66f from the respective input module 14, the controller 12, and the output module 16. If the configurations of messages 66a, 66b and 66c exactly match (after complementing) configuration data of messages 66d, 66e and 66f, the configuration was successful.

The configuration data may be shown to a human operator for confirmation. If the operator finds that the configuration is correct, the configuration is applied as indicated by process 68 shown in Fig. 10 through messages 68a, 68b and 68c from the configuration terminal 24 to the respective input module 14, the controller 12, and the output module 16. The devices must acknowledge these messages via messages 68d, 68e and 68f within a predetermined time interval or the configuration will be cleared and no configuration will be considered to have occurred. The configuration data of messages 66 and 68 may be sent using only the standard network protocol 56.

Once the configuration is complete, the safety protocol enters a start-up phase shown generally in Figs. 8 and 10. During the start-up phase, the necessary safety connections are established and the configuration data is used to verify that the

connections expected are those which are in fact made. The purpose of the start-up portion of the configuration is to prevent erroneous connections from being opened between: (1) devices in the safety system and other erroneous devices in the safety system, and (2) devices in the safety system and other devices not in the safety system in a mixed system.

In this start-up process, indicated by process block 70 of Fig. 10, the connections are confirmed from the controller 12 to the input module 14 and the output module 16. In particular, the producers 80 in controller 12 (shown in Fig. 9) send out open connection messages 70a and 70b to the input module 14 and the output module 16, respectively. The appropriate consumers 82 respond with connection acknowledgment message 70c and 70d, respectively. The producers 80 in controller 12 and input module 14 then send the configuration data to the consumer 82 in the controller 12 as indicated by messages 70e and 70f. The controller's consumers 82 check to see that the configuration data matches their configuration data and then send acknowledgment messages 70f and 70g acknowledging that match. At messages 72a and 72b, conventional I/O data may then commence to be sent.

Referring again to Fig. 10, the data 72a and 72b will be transmitted according to the portions of the safety protocol indicated by process blocks 72 involving formation of the safety message 60 incorporating safety error detection data 58 into the network message 61 as has been described above, and according to message handling protocols 74 operating independent of and in conjunction with the content of the safety message 60 which will now be discussed.

Message Handling Safety Protocols

(1) Normal Transmission

Referring generally to Figs. 10 and 11, the message handling protocols 74 provide for message time measurements and respond to errors in the safety error detection data 58 during run-time. These message-handling protocols 74 are implemented in the safety protocol circuits 32, 40 and 46 or may be implemented in software and are different for producers and consumers.

Referring now to Figs. 11 and 19 for a normal, run-time transmission, the producer 80 upon run-time will send safety messages 84 (encapsulated in the standard network message 61 per safety message 60 as has been described above) to the consumer 82 per Fig. 11. This sending is indicated generally in Fig. 19. Immediately prior to sending the message 84, a periodic timer is started per process block 89 and a reply timer

is started at the moment the message 84 is transmitted per process block 91. The periodic timer interval 86 is longer than the reply timer interval 88 as set in the configuration process described above.

Referring now to Fig. 9, 11 and 20, the consumer 82 prior to receiving the message 84 is continually checking to see if the periodic time interval 86' of its own periodic timer (started at the consumer's receipt of the last message 84) has expired as indicated in decision block 92. The periodic timer value 86' is generally identical to periodic timer value 86.

If the periodic timer has expired, a failure is indicated and the program proceeds to process block 134, a safety state, as will be described below.

If timer value 86 has not expired, then at decision block 90, the consumer 82 checks to see if the message 84 has arrived. If no message 84 has arrived the program proceeds back to decision block 92 to again check if the periodic timer 86 has expired.

Assuming that a message 84 has arrived prior to expiration of the periodic timer 86, then the program proceeds to decision block 112 to check the CRC of the message 84.

Assuming that the CRC is correct, the program proceeds to decision block 96 checks to make sure that the sequence count is one greater than the sequence count of the last message received.

If the sequence count is correct, then the program proceeds to process block 94 and the periodic timer 86 is reset. At process block 95, the data is applied, for example, to an output or to update variables, and then at process block 98, an acknowledgement message100 is returned to the producer 80.

Referring again to Fig. 19, the producer 80 receiving the acknowledge message at decision block 102, proceeds to decision block 106 to determine if the periodic timer 86 has expired.

Assuming that the periodic timer has not expired, the program proceeds to decision block 124 to check the CRC of the acknowledgement message 100. The cyclic redundancy code should match the data of the safety message 60 transmitted.

Again, assuming that the CRC is correct, the program proceeds to decision block 125 to determine whether the sequence count of the acknowledgment message 100 matches that of the message 84 that was sent.

If so, then at decision block 127, the data sent in message 84 is compared to the data of the acknowledgement message 100. If the is a match, then the program proceeds

to decision block 129 where it loops until the periodic timer has expired, and then proceeds to process block 110 to prepare a new message 84.

This process is repeated for multiple transmissions of safety messages 84 and acknowledgement messages 100.

(2) Message Received but Corrupted

Referring now to Fig. 11 in one potential error the safety message 84 is corrupted for example by electromagnetic interference 85. In this case a message is received at the consumer 82, as indicated by Fig. 20 per process block 90, within the periodic timer value 86' as measured by process block 92 however there is an error in the CRC data as determined by decision block 112. In this case, the program proceeds to process block 114 and no action is taken and in particular no acknowledgement message 100 is returned.

Referring to Fig. 19, in this case there will be no acknowledgment message 100 received by the producer 80 at process block 102. The program proceeds to decision block 116 to determine if the periodic time interval 86has expired. If so, the failure is indicated and the program proceeds to the safety state of process block 126.

If the periodic timer interval 86 has not expired, the program will proceed to decision block 118 to see if the shorter reply timer interval 88 has expired. If not, the program will loop back to process block 102. If so, the program will proceed to process block 120 to check if the retry limit has been exceeded. Initially this may not be the case and the program will proceed to process block 122 and a repeat message 84' having the same sequence count will be sent at process block 84, as also indicated by Fig. 12. If the retry limit has been exceeded, the program proceeds to the safety state 126

This repeat message 84' will be received at the consumer 82 as indicated by process block 90 of Fig. 20 and assuming that it is correct it and that it has arrived within the periodic timer interval 86' based on the previous non-erroneous message, this message 84' results in the sending of an acknowledgment message 100 at process block 98 per the steps described above.

Typically, if only one missed transmission has occurred, the acknowledgment message 100 will occur within the periodic timer interval 86 of the producer and messages will be continued to be exchanged normally as has been previously described with respect to Fig. 11.

(3) Message Not Received

Referring now to Fig. 13, in the previous example, the safety message 84 arrived at the consumer 82 to be detected, albeit with errors. It is possible that the safety message 84 will not arrive at the consumer 82 either as a result of such extreme interference that it is not recognizable as a message under low level network protocols, or as a result of component failures between the producer and the consumer of an intermittent nature. Under this situation, the producer 80 sends the message 84 but the consumer does not receive a message at process block 90 of Fig. 20.

The "no action" block 114 of Fig. 20 of the consumer (as described above) is thus not encountered but the result is in any case the same: the consumer 82 takes no action.

Thus, as described previously with respect to Fig. 12 at the expiration of the reply timer at the producer 80, the producer 80 will produce a second message 84' which if received will result in an acknowledgment message 100 initiating a string of normal communications.

(4) Acknowledgement Message Received but Corrupted

Referring now to Fig. 14 the safety message 84 may successfully reach the consumer 82 with no errors but the acknowledgement message 100 may have errors introduced by electromagnetic interference 85. In this case the producer 80 reacts as shown in Fig. 19 by decision block 106 to detect a receipt of an acknowledgment message 100 within the periodic timer interval 86. But there is an error in the data of the acknowledgment message 100.

If the CRC is correct as determined by decision block 124 and it is the sequence count that is wrong per process block 124, then the program enters the safety state 126 in which outputs and inputs of the consumer 82 are set to a predefined safety state of the configuration data. Similarly, if the sequence count is correct but the acknowledgement data does not match per decision block 127, the program proceeds to the safety state 126. If the consumer 82 is the controller 12 messages may be sent to other I/O devices, indicated in the configuration data signaling them to move to the safety state as well.

Assuming at process block 124 that the CRC code does not match the safety message 60. indicating a corruption in the safety message rather than an erroneous duplicate message, the program proceeds to decision block 118 to see if the reply timer has expired as described above. When the reply timer expires the program proceeds to process block 120 as described above and checks the retry counter to see if the retry limit has been exceeded. If so, the program proceeds to the safety state 126 however often this

will not have occurred and the program proceeds to process block 122 and a retry message 84' is prepared as indicated in Fig. 14.

Assuming this retry message evokes a non-corrupted acknowledgment message 100' communication continues in normal fashion.

(5) Acknowledgment Message Not Received

Referring now to Fig. 15 it is possible that the acknowledgment message 100 is lost completely either through interference or temporary communication failure. In that case, as has been previously described, a duplicate message 84 will be sent from the producer 80 however the sequence count will be identical to the sequence count of a message 84 previously received by the consumer 82. In this case as shown in Fig. 20 at process block 112 the CRC will be correct but as tested at subsequent decision block 96 the sequence code will be wrong. The program, in this case, proceeds to process block 130 to check if the sequence code is one less than that expected. If not the program proceeds to the safety state 134. If so, however, the consumer 82 must conclude that the acknowledgment message 100 was lost and an acknowledgment of the previous message is sent again by the consumer at process block 132.

(6) No Messages Received

Finally as shown in Fig. 15 the producer may be unable to connect with the consumer within the periodic interval 86' of the consumer. In that case the program proceeds to the safety state 134 directly from decision block 92 as shown in Fig. 20

Reduction of Network Induced Skew

Referring now to Figs. 2, 17 and 18 duplicate input signals 18a and 18b must be filtered to accommodate differences in their sampling of their associated input circuits 28a and 28b in input module 14. As shown in Fig. 17, the signal on line 18a may be sampled at intervals 140a (shown by dotted lines) whereas input signal on line 18b (possibly identical to that of 18a) may be sampled at intervals 140b shown by solid lines This difference in sampling rates can mean at a time t_0 that signal 18b' processed by interface circuitry 28a is in a high state whereas signal 18a as processed by interface circuitry 28a is in a high state. This skew signals 18a' and 18b' may also be caused by slight mechanical delays in dual contact systems.

If not corrected, this skew can produce a momentary erroneous state between time t_0 and t_1 suggesting a failure. These false error indications must be eliminated and this is typically done by establishing a window 150 during which lack of coincidences of the signals 18a and 18b will be ignored.

In a network system this may involve sending the signals 18a' and 18b' to the respective processors 42a and 42b via network messages. Skew in the transmission of these messages resulting from their queuing for transmission on the network 15 may require that the window 150 be increased. An additional problem in such an approach arises in the fact that the windowing process requires repeated samples of each input 18a and 18b. For example, upon a change in signal 18b (where there is lack of coincidence with signal 18a) it is necessary to continue to monitor (and transmit) the subsequent values of signal 18a within the window 150. Accordingly the network 15 is taxed with the additional transmission of multiple samples of each input. As the need for high response times increases the number of samples, network traffic also increases.

Accordingly in the present invention, using the configuration shown in Fig. 2, the microcontroller 30a and 30b communicate over bus 34 to resolve any lack of coincidence, using window filtering to determine a common value of 18a and 18b and to send this common value in separate messages in a "prefiltered state" to the processors 42. Because the processors 42 operate synchronously no additional skew is introduced or need be eliminated. As a result the skew window is much reduced and network traffic is much reduced.

In particular, referring to Fig. 18, the microcontroller 30a and 30b operate a filter program testing at decision block 142 whether two inputs are coincident. This may means that the two inputs (e.g., signals 18a and 18b) are the same or one is the inversion of the other according to a predetermined convention. If there is coincidence, then the output state of those inputs is set to their coincident value as indicated by process block 144. This corresponds generally to region 146 of Fig. 17.

The common output of process block 144 may be periodically transmitted or transmitted only at an instant of change of state.

If no coincidence occurs as indicated by region 148 of Fig. 17 then the program proceeds to process block 145 and samples within a window 150 for example starting at the present and proceeding forward two samples to see if coincidence can be found. If coincidence can be found within the window 150 as indicated by process block 152 then the window coincidence value is used as the output as indicated by process block 154.

If no coincidence can be found then the program proceeds to the safety state 136.

The microcontrollers 30a and 30b may further filter the input data by mapping a large number of inputs states into a lesser number of transmission states. These fewer numbers of transmission states (such as may be represented by less transmitted data),

reduce the burden imposed on network bandwidth by high reliability systems. Further, in the case where only changes in transmission states are transmitted, the fewer number of states will cause fewer transmissions. Thus, for example, two inputs, such as from a dual button safety lockout, may be resolved to two states, "run" or "stop". The two bits of input data may be filtered to one bit of transmitted data. This compression may operate in conjunction with the windowing described above.

The above description has been that of a preferred embodiment of the present invention, it will occur to those that practice the art that many modifications may be made without departing from the spirit and scope of the invention. In order to apprise the public of the various embodiments that may fall within the scope of the invention, the following claims are made.

CLAIMS

WE CLAIM:

1. A network-independent, high-reliability communications system for an industrial controller using a standard serial communications network, the communications system comprising:

a first I/O communications circuit receiving I/O data for control of an industrial process;

a first network-independent protocol device receiving the I/O data and formatting it for transmission under a network-independent protocol to produce high-reliability formatted data formatted to reduce undetected transmission loop errors;

a first standard network protocol device receiving the high-reliability formatted data and further formatting it for transmission under a protocol of the standard serial communications network, to produce doubly-formatted data for transmission on the standard serial communications network, the protocol of the standard serial communications network also formatting data to reduce undetected transmission loop errors;

a second standard network-protocol device receiving the doubly-formatted data from the standard serial communications network and extracting the high-reliability formatted data according to the protocol of the standard serial communications network;

a second network-independent protocol device receiving the high-reliability formatted data and extracting the I/O data; and

a second I/O communications circuit receiving I/O data for control of an industrial process from the second network-independent protocol device;

whereby high-reliability transmissions may be simply obtained on an arbitrary standard serial communications network protocol.

2. The industrial controller of claim 1 wherein the first and second I/O communications circuits are selected from the group consisting of an industrial controller, an input circuit for an industrial controller, a bridge, and an output circuit for an industrial controller.

3. The industrial controller of claim 1 wherein the first network-independent protocol device formats the I/O data by adding error detection data consisting of: a cyclic redundancy code related to the I/O data and a sequence count related to a local order of transmission of the I/O data with respect to other I/O data being transmitted.

4. The industrial controller of claim 1 wherein the second network-independent protocol device further generates an acknowledgment message upon receipt of the I/O data and formats it under the network-independent protocol to produce a high-reliability formatted acknowledgment data;

and wherein the second standard network protocol device receives the high-reliability formatted acknowledgment data and further formats it for transmission under the protocol of the standard serial communications network, to produce doubly-formatted acknowledgment data for transmission on the standard serial communications network;

and wherein the first standard network-protocol device receiving the doubly-formatted acknowledgment data from the standard serial communications network and extracts the high-reliability formatted acknowledgment data according to the protocol of the standard serial communications network;

and wherein the first network-independent protocol device receiving the high-reliability formatted acknowledgment data checks the data to detect transmission loop errors.

5. The industrial controller of claim 4 wherein the acknowledgment data includes the I/O data and the first network-independent protocol device detects errors by comparing the I/O data to the acknowledgment data.

6. The industrial controller of claim 1 wherein the first network-independent protocol device operates to start a timer upon receipt of the I/O data and wherein the first network-independent protocol device detects errors by checking a time on the timer against an allowable time upon receipt of the acknowledgment message.

7. The industrial controller of claim 1 wherein the first network-independent protocol device transmits I/O data on a regular interval and wherein the second network-independent protocol device detects errors by comparing the time at which the last I/O data was received against the time interval.

8. The industrial controller of claim 1 wherein the second network-independent protocol device evaluates the high-reliability formatted data to detect transmission loop errors of the I/O data and upon the detection of an error for I/O data assume a default safety state of the I/O data.

9. The industrial controller of claim 4 wherein the first network-independent protocol device evaluates the high-reliability formatted data to detect transmission loop errors of the I/O data and upon the detection of an error for I/O data assume a default safety state of the I/O data.

10. The industrial controller of claim 1 wherein the standard serial communications network is selected from the group of networks consisting of Ethernet, DeviceNet, ControlNet, Fire Wire and Field Bus.

11. A method of establishing high reliability communication among components of an industrial control system exchanging control signals with a controlled process, the components communicating over a standard network, the method comprising the steps of:

(a) transmitting a configuration message from a configuration source to a first component and a second component over the standard network using a standard network protocol, the configuration message providing data related to a high reliability communications protocol usable on the standard network;

(b) receiving at the configuration source a configuration response message from the first component and the second component, the configuration response message describing data of a configuration message previously received by the first component and the second component; and

(c) allowing the communication of control signals between the first and second component, as defined by the data of the configuration message, only if the configuration response message received by the configuration source describes the same data as the configuration message transmitted from the configuration source.

12. The method of claim 11 wherein at step (a) the data of the configuration message is stored at the first component and the second component and including the further step of:

(d) if the configuration response message received by the configuration source describes different data from the data of the configuration message, sending a clear message from the configuration source to the first component and the second component causing the clearing of the configuration message stored at the first component and the second component.

13. The method of claim 11 wherein the configuration response messages describes data by sending a ones complement transformation of the data in the configuration message.

14. The method of claim 11 further including the steps of:

(e) when the configuration response message received by the configuration source describes the data of the configuration message transmitted by the configuration source, communicating a configuration apply message from the configuration source to the first component and the second component; and wherein

at step (c) communicating control signals between the first component and the second component using the high reliability communication protocol on the standard network only if the apply message is received by the first component and the second component.

15. The method of claim 14 wherein at step (a) the data of the configuration message is stored at the first component and the second component and including the further step of:

(d) if the apply message is not received by the first component and the second component within a predetermined time after the receipt of the configuration message, clearing of the configuration message stored in at least one of the first component and second component.

16. The method of claim 14 further including the steps of:

(f) when the configuration apply message is received by the first component and the second component, communicating an apply acknowledgement message from the first component and the second component to the configuration source; and wherein

at step (c) communicating control signals between the first and second components only if the apply acknowledgment message is received by the configuration source.

17. The method of claim 11 wherein the data of the configuration message includes data uniquely identifying the first and second components as parties of communication on the standard network.

18. The method of claim 17 wherein the configuration data is selected from the group consisting of: a serial number of the first component, a serial number of the second component, a device type identifying the functional type of the first component, a device type identifying a functional type of the second component, a vendor identification of the first component, a vendor identification of the second device, a produce code used by the vendor to identify the first component, a product code used by the vendor to identify the second component, a revision number of programming of the first component, a revision number of programming of the second component.

19. The method of claim 11 wherein the data of the configuration message includes data defining parameters of operation of the high reliability protocol for communication between the first and second components.

20. The method of claim 19 wherein the configuration data is selected from the group consisting of: a periodic time interval indicating a minimum expected frequency of initiation of transmission of data between the first and second components, a reply timer interval indicating a maximum expected delay between an initiation of transmission and a reply to that transmission between the first and second components, a filter count indicating a window of time within a coincidence of redundant control signals must exist for no error condition to occur, a retry limit indicating how many transmission message retries are allowed before a error condition occurs, a safety state to which outputs of the first or second components will revert to upon an error condition, and an I/O family

indicating other output of the first or second components which should revert to a safety state upon an error condition related to a single output of the first or second device.

21. The method of claim 11 including the further steps of:

(d) opening a connection on the standard network between the first and second component;

(e) sending a message from the first component to the second component using the standard communication protocol and identifying the data of the configuration message;

(f) only if the data of the configuration message received at the second component matches configuration data previously received from the configuration source and stored at the first and second component, sending an acknowledgement signal from the second component to the first component; and wherein

at step (c) allowing the communication of control signals between the first and second component, as defined by the data of the configuration message, only after receipt of the acknowledgement signal

22. The method of claim 21 wherein the data of the configuration message includes data uniquely identifying the first and second components as parties of communication on the standard network.

23. The method of claim 22 wherein the configuration data is selected from the group consisting of: a serial number of the first component, a serial number of the second component, a device type identifying the functional type of the first component, a device type identifying a functional type of the second component, a vendor identification of the first component, a vendor identification of the second device, a produce code used by the vendor to identify the first component, a product code used by the vendor to identify the second component, a revision number of programming of the first component, a revision number of programming of the second component.

24. The method of claim 21 wherein the data of the configuration message includes data defining parameters of operation of the high reliability protocol for communication between the first and second components.

25. The method of claim 24 wherein the configuration data is selected from the group consisting of: a periodic time interval indicating a minimum expected frequency of initiation of transmission of data between the first and second components, a reply timer interval indicating a maximum expected delay between an initiation of transmission and a reply to that transmission between the first and second components, a filter count indicating a window of time within a coincidence of redundant control signals must exist for no error condition to occur, a retry limit indicating how many transmission message retries are allowed before a error condition occurs, a safety state to which outputs of the first or second components will revert to upon an error condition, and an I/O family indicating other output of the first or second controllers which should revert to a safety state upon an error condition related to a single output of the first or second device.

26. A method of establishing high reliability communication among components of an industrial controller some of which receive control signals from a controlled process, the components communicating over a standard network, the method comprising the steps of:

(a) establishing at least two redundant logical message producers associated with a given received control signal;

(b) opening a logical connection between each of the two logical message producers and two corresponding logical message consumers;

(c) transmitting data including a given received control signal on the connections from the logical message producers to the logical message consumers;

(d) after receipt of the uncorrupted data at each logical message consumer, transmitting reply data including the given received control signal on the connection to the logical message producers; and

(e) responding at the logical message producers to an absence of an uncorrupted receipt of a transmission of reply data subsequent to step (c) on the connection by entering a predetermined safety state.

27. The method of claim 26 including the further steps of:

(f) comparing uncorrupted reply data subsequent to step (c) between the two logical message producers; and

at step (e) further responding to a failure of the reply data of step (f) to match by causing the logical message producers to enter the predetermined safety state.

28. The method of claim 26 wherein the determination of whether data is uncorrupted at step (d) uses a cyclic redundancy code incorporated into the data and a function of the received control signal.

29. The method of claim 26 wherein the determination of whether data is uncorrupted at step (d) uses a message sequence count to indicate a relative order of messages holding the transmitted data.

30. The method of claim 26 including the further steps of:

(f) comparing uncorrupted data subsequent to step (b) between the two logical message consumers; and

at step (e) further responding to a failure of the data of step (f) to match by causing the logical message consumers to enter the predetermined safety state.

31. The method of claim 30 wherein the two logical message producers are established in a single physical device having one connection to a standard serial network.

32. The method of claim 26 wherein the two logical message producers are established in two separate physical devices having two physical connections to a standard serial network.

33. The method of claim 26 wherein the two logical message consumers are established in a single physical device having one connection to a standard serial network.

34. The method of claim 26 wherein the two logical message consumers are established in two separate physical devices having one connection to a standard serial network.

35. The method of claim 26 wherein the data and reply data are transmitted using protocols for a network selected from the group consisting of Ethernet, DeviceNet, ControlNet, Firewire or FieldBus.

36. The method of claim 26 including the further step of:

(f) responding a predetermined number of times at the logical message producers to a failure to receive an uncorrupted receipt of a transmission of reply data subsequent to step (c) but within a predetermined reply time by repeating step (c).

37. The method of claim 26 wherein the two logical message producers are part of an industrial controller and wherein step (e) further includes the step of transmitting a signal to the two logical message consumers instructing them to enter a predetermined safety state.

38. A high reliability industrial control system comprising:

a controller including a first network interface to a shared serial network;

an input module having at least two interface circuits for receiving at least two redundant input signals, the interface circuits communicating with at least one processor via an internal bus, the processor further communicating with a second network interface to the shared serial network;

wherein the processor executes a stored program to:

(i) receive the redundant input signals processed by the interface circuits;

(ii) determine a coincidence of the redundant input signals within a window of a predefined time period; and

(iii) only when there is coincidence within the window, transmitting via the second network interface, at least one coincidence signal indicating a coincident state of the redundant input signals to the controller.

39. The industrial control system of claim 38 wherein the processor executes the stored program to transmit to the controller at least two redundant messages on the shared network indicating the coincident state of the redundant input signals when there is coincidence within the window.

40. The industrial control system of claim 38 wherein the interface circuit includes two processors with each interface circuit communicating with a different processor, and

wherein the processors communicate with each other via an internal bus and execute stored programs to:

(i) each receive a different of the redundant input signals processed by the interface circuits;

(ii) communicate with the other processor to determine a coincidence of the redundant input signals within a window of a predefined time period; and

(iii) only when there is coincidence within the window, each processor transmitting to the controller via the second network interface, a common coincidence signal indicating a coincident state of the redundant input signals.

41. The industrial control system of claim 40 wherein the second network interface includes two redundant interface circuits each dedicated to one of the processors.

42. The industrial control system of claim 38 wherein the processor executes the stored program to determine a coincidence as existing when one related signal is the complement of the other related signal within the window.

43. The industrial control system of claim 38 wherein the processor executes the stored program to determine a coincidence as existing when one related signal is the same logical state as the other related signal within the window.

44. The industrial control system of claim 38 wherein the input circuits sample the redundant input signal at regular sample times and wherein the processor executes the stored program to determine a coincidence as existing within the window by detecting a lack of coincidence and reviewing a predetermined number of samples commensurate with the period of time of the window and determining a coincidence only if coincidence is obtained at one of the predetermined number of samples.

45. The industrial control system of claim 38 including further:

an output circuit having a third network interface to the shared serial network for creating an output signal related to at least one of the redundant input signals and wherein the output circuit communicates its output signal to the input module via the third network interface and wherein the communicated output signal is the coincidence signal.

46. The industrial control system of claim 38 wherein the processor executes the stored program to only when there is no coincidence followed by coincidence within the window, transmitting via the second network interface, at least one coincidence signal indicating a coincident state of the redundant input signals to the controller.

47. The industrial control system of claim 38 wherein the processor further executes the stored program to, only when there is no coincidence within the window, enter a safety state indicating failure of the industrial control system.

48. The industrial control system of claim 47 wherein the processor executes the stored program to transmit via the second network interface on a regular basis one of a coincidence signal indicating a coincident state of the redundant input signals to the controller or a safety state signal indicating a failure of the industrial control system.

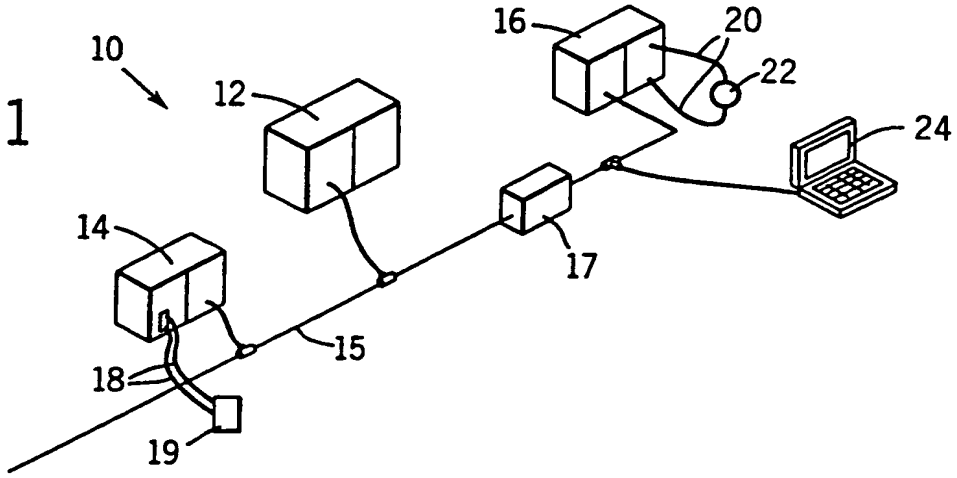
49. The industrial control system of claim 38 wherein the input module includes at least four input circuits for receiving at least two pairs of redundant input signals and wherein the processor further executes the stored program to:

(i) receive the two pairs of redundant input signals processed by the interface circuits;

(ii) determine a first and second coincidence of the respective pairs of redundant input signals within at least one window of the predefined time period; and

(iii) only when there is coincidence within the window for each of the two pairs of input signals, map the state of the two pairs of inputs to a lesser number of transmission states, transmitting via the second network interface, at least one coincidence signal indicating a transmission state of the redundant input signals to the controller.

FIG. 1



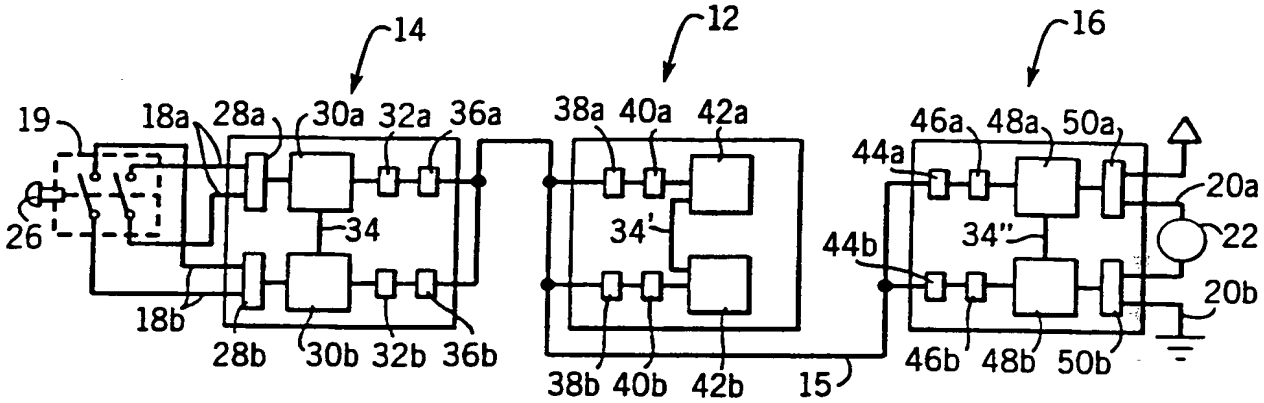


FIG. 2

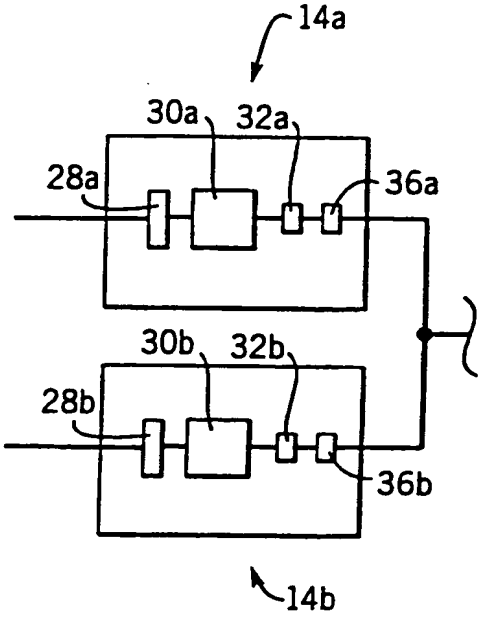


FIG. 3

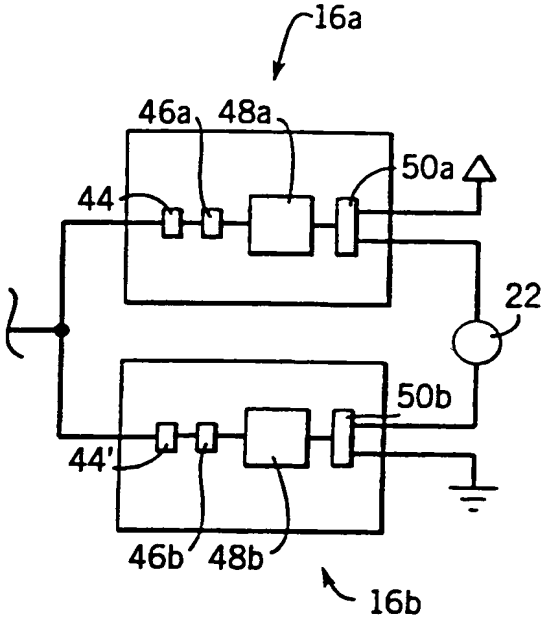


FIG. 4

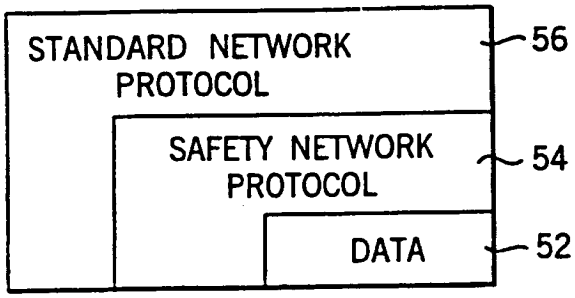


FIG. 5

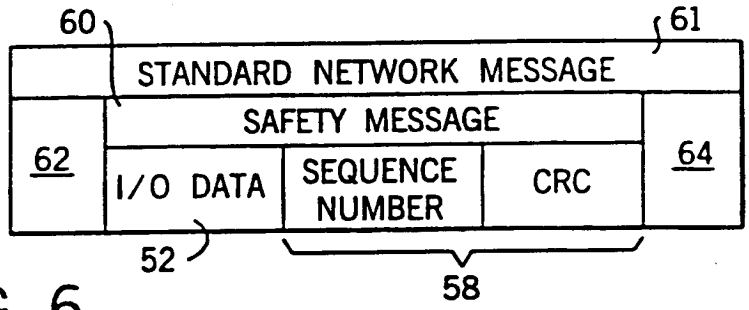


FIG. 6

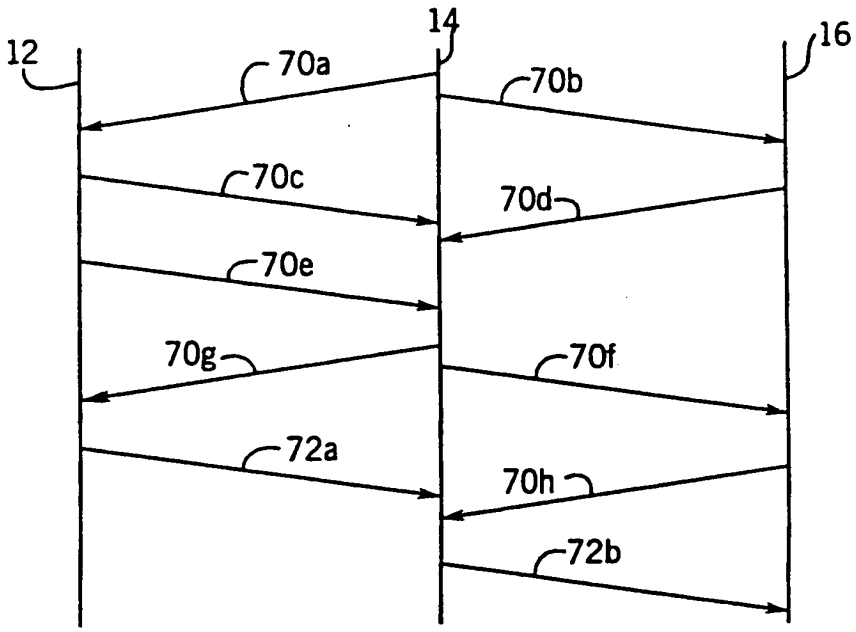
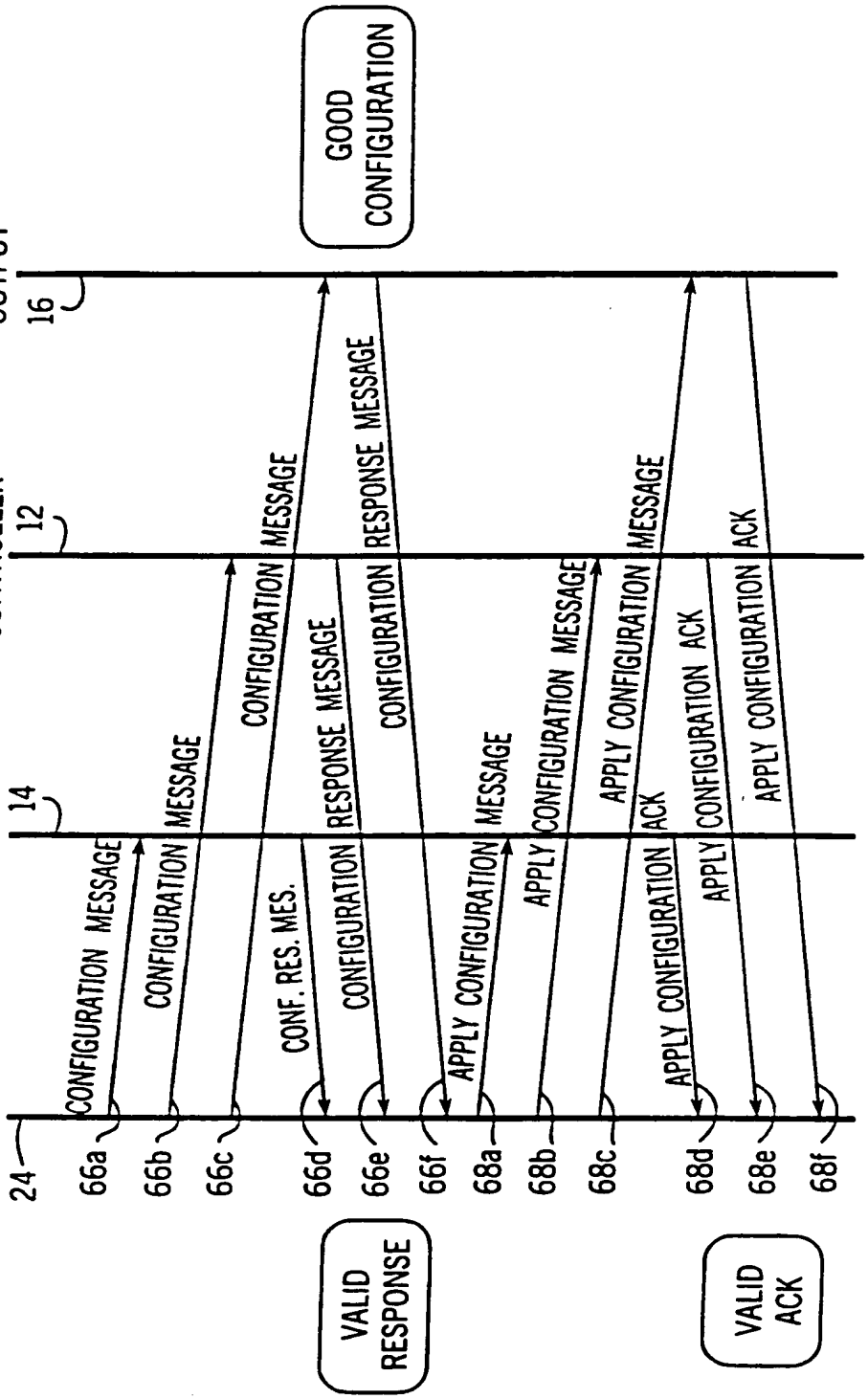


FIG. 8

FIG. 7


SAFETY CONFIGURATION TERMINAL
24
INPUT 14
CONTROLLER 12
OUTPUT 16
CONFIGURATION MESSAGE
CONFIGURATION MESSAGE
CONFIGURATION MESSAGE
CONF. RES. MES.
CONFIGURATION RESPONSE MESSAGE
CONFIGURATION RESPONSE MESSAGE
APPLY CONFIGURATION MESSAGE
APPLY CONFIGURATION MESSAGE
APPLY CONFIGURATION MESSAGE
APPLY CONFIGURATION ACK
APPLY CONFIGURATION ACK
APPLY CONFIGURATION ACK
66a
66b
66c
66d
66e
66f
68a
68b
68c
68d
68e
68f
GOOD CONFIGURATION
VALID RESPONSE
VALID ACK

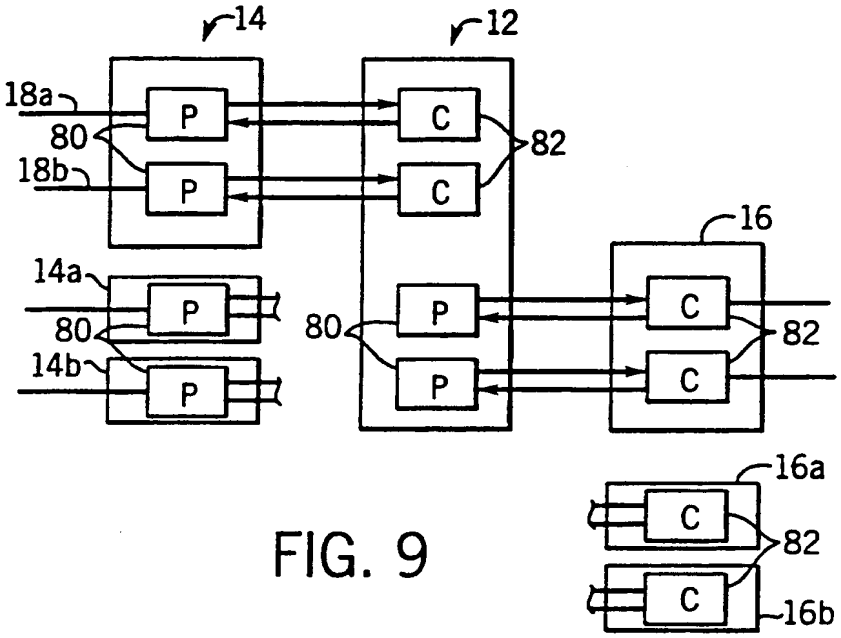

14
12
18a
80
18b
P
C
82
16
14a
14b
16a
16b

FIG. 9

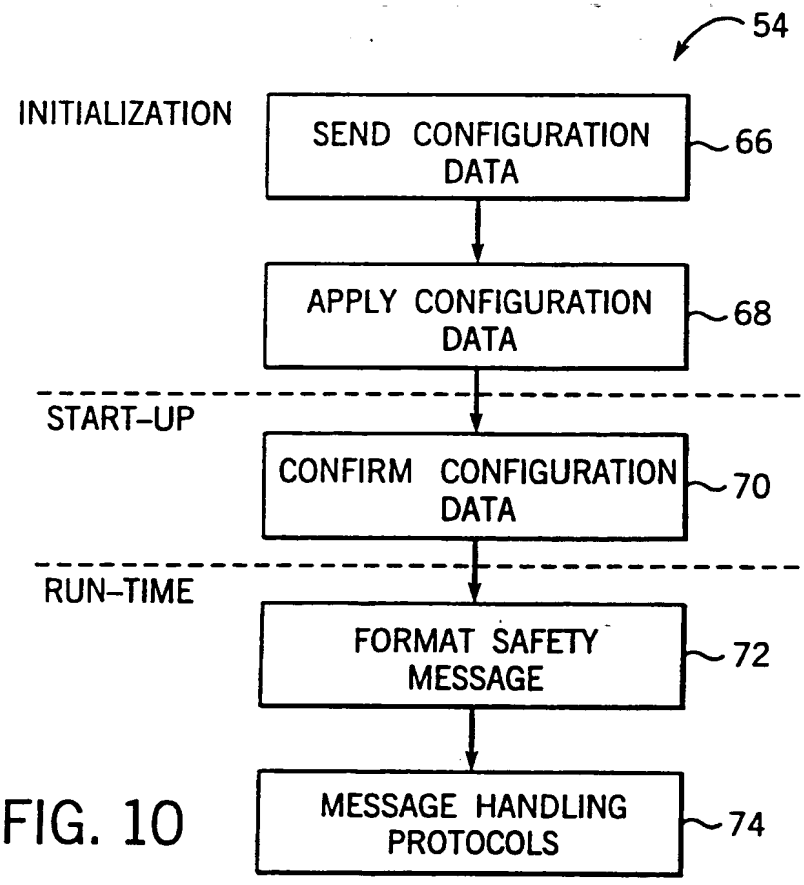

54
INITIALIZATION
SEND CONFIGURATION DATA
66
APPLY CONFIGURATION DATA
68
START-UP
CONFIRM CONFIGURATION DATA
70
RUN-TIME
FORMAT SAFETY MESSAGE
72
MESSAGE HANDLING PROTOCOLS
74

FIG. 10

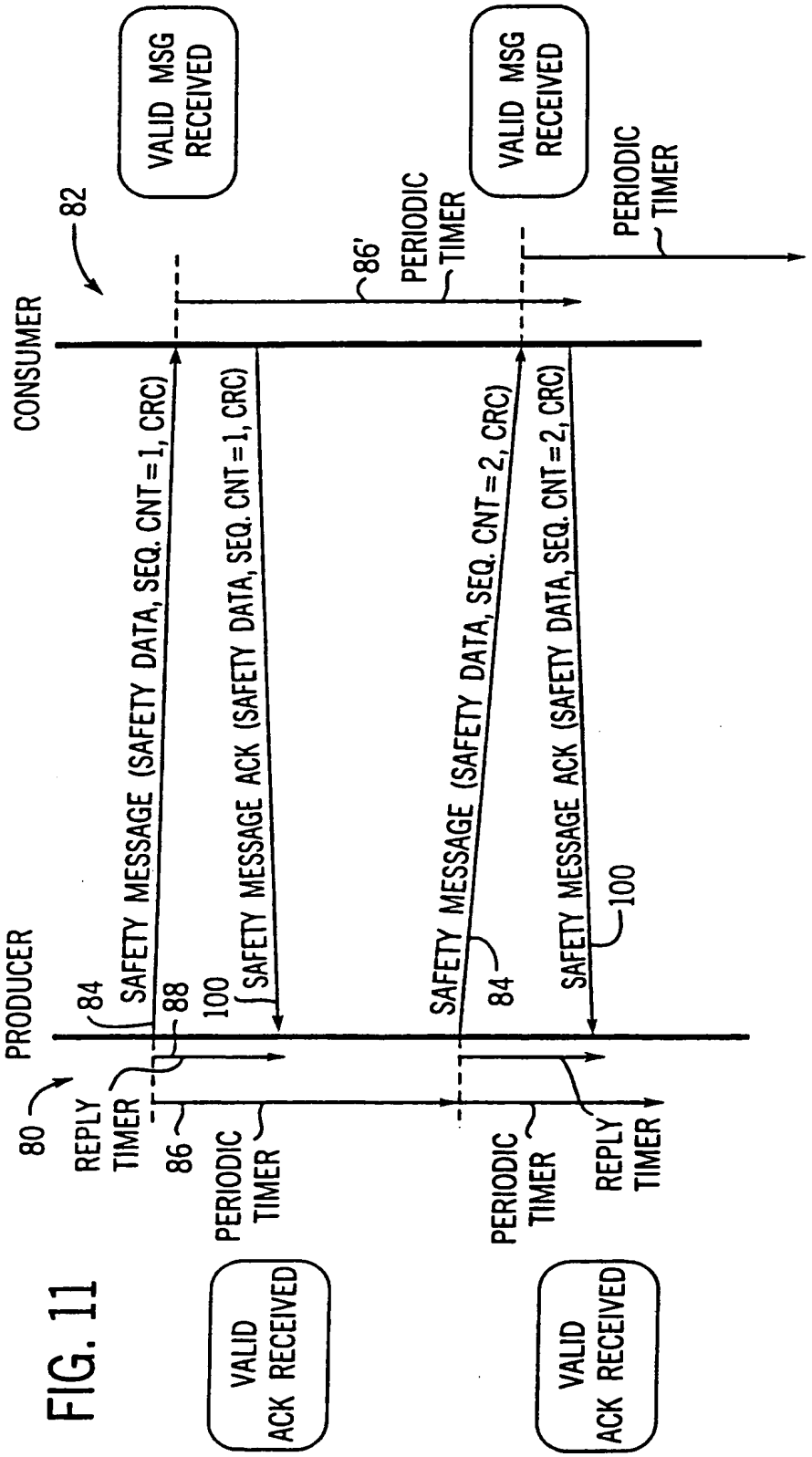

PRODUCER
CONSUMER
82
80
84
88
100
86
86'
REPLY TIMER
PERIODIC TIMER
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=1, CRC)
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=2, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=2, CRC)
VALID ACK RECEIVED
VALID MSG RECEIVED

FIG. 11

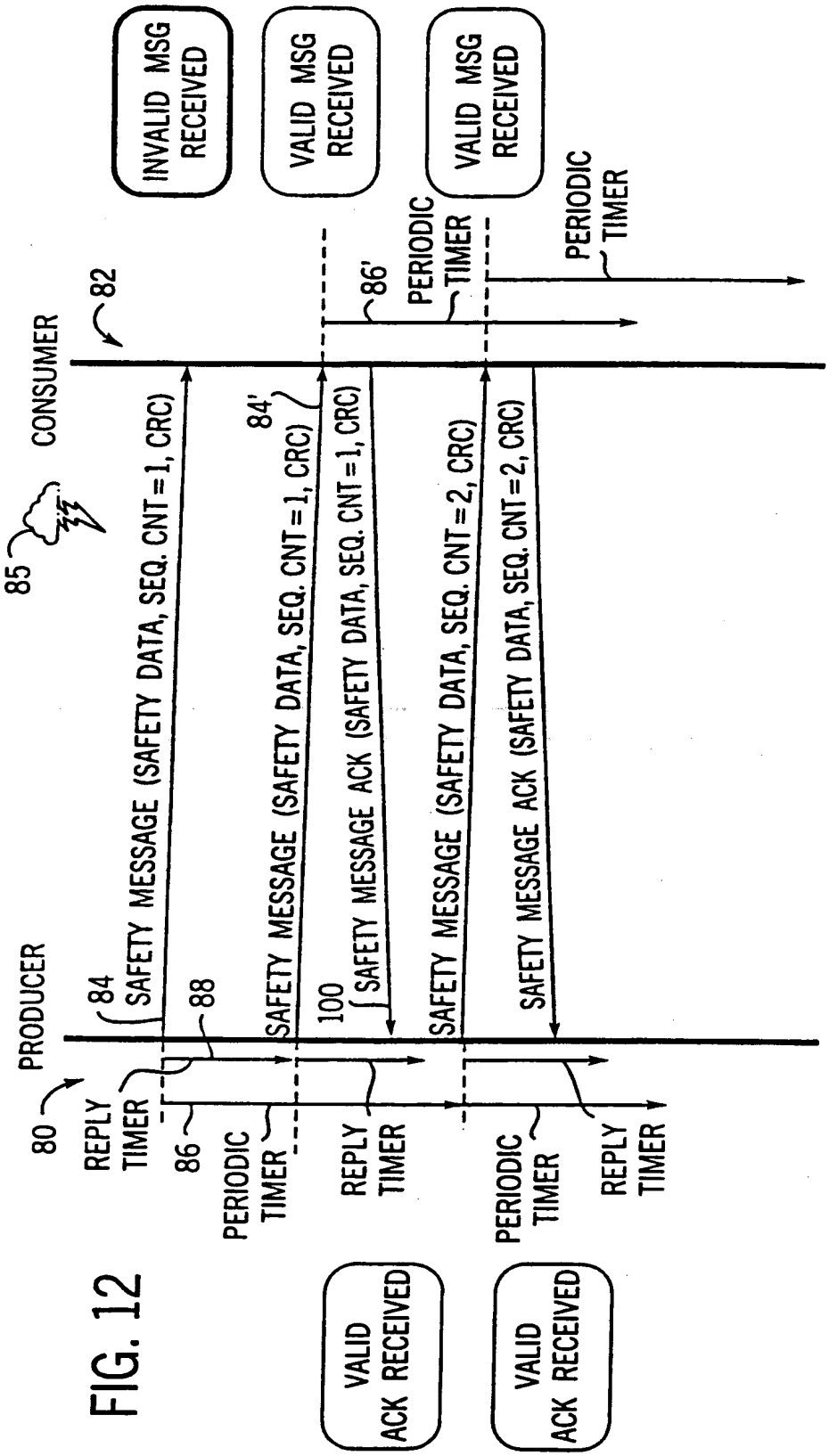

PRODUCER
CONSUMER
80
82
85
84 SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
88
REPLY TIMER
86
PERIODIC TIMER
84' SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
100 SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=1, CRC)
REPLY TIMER
86' PERIODIC TIMER
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=2, CRC)
PERIODIC TIMER
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=2, CRC)
REPLY TIMER
INVALID MSG RECEIVED
VALID MSG RECEIVED
VALID MSG RECEIVED
VALID ACK RECEIVED
VALID ACK RECEIVED

FIG. 12

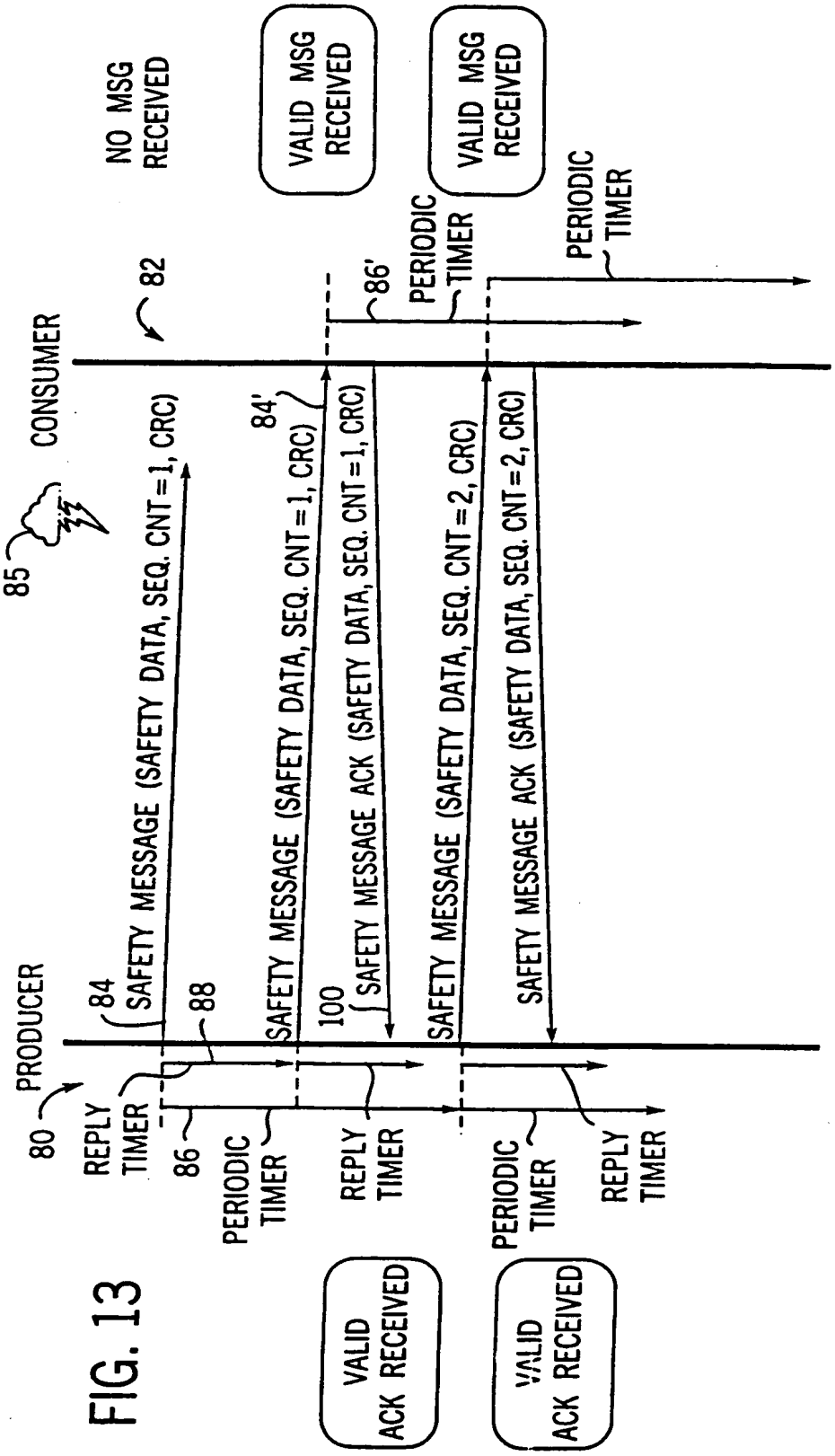

PRODUCER
CONSUMER
80
82
85
84
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
88
84'
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
100
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=1, CRC)
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=2, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=2, CRC)
REPLY TIMER
86
PERIODIC TIMER
REPLY TIMER
PERIODIC TIMER
REPLY TIMER
NO MSG RECEIVED
VALID MSG RECEIVED
86'
PERIODIC TIMER
VALID MSG RECEIVED
PERIODIC TIMER
VALID ACK RECEIVED
VALID ACK RECEIVED

FIG. 13

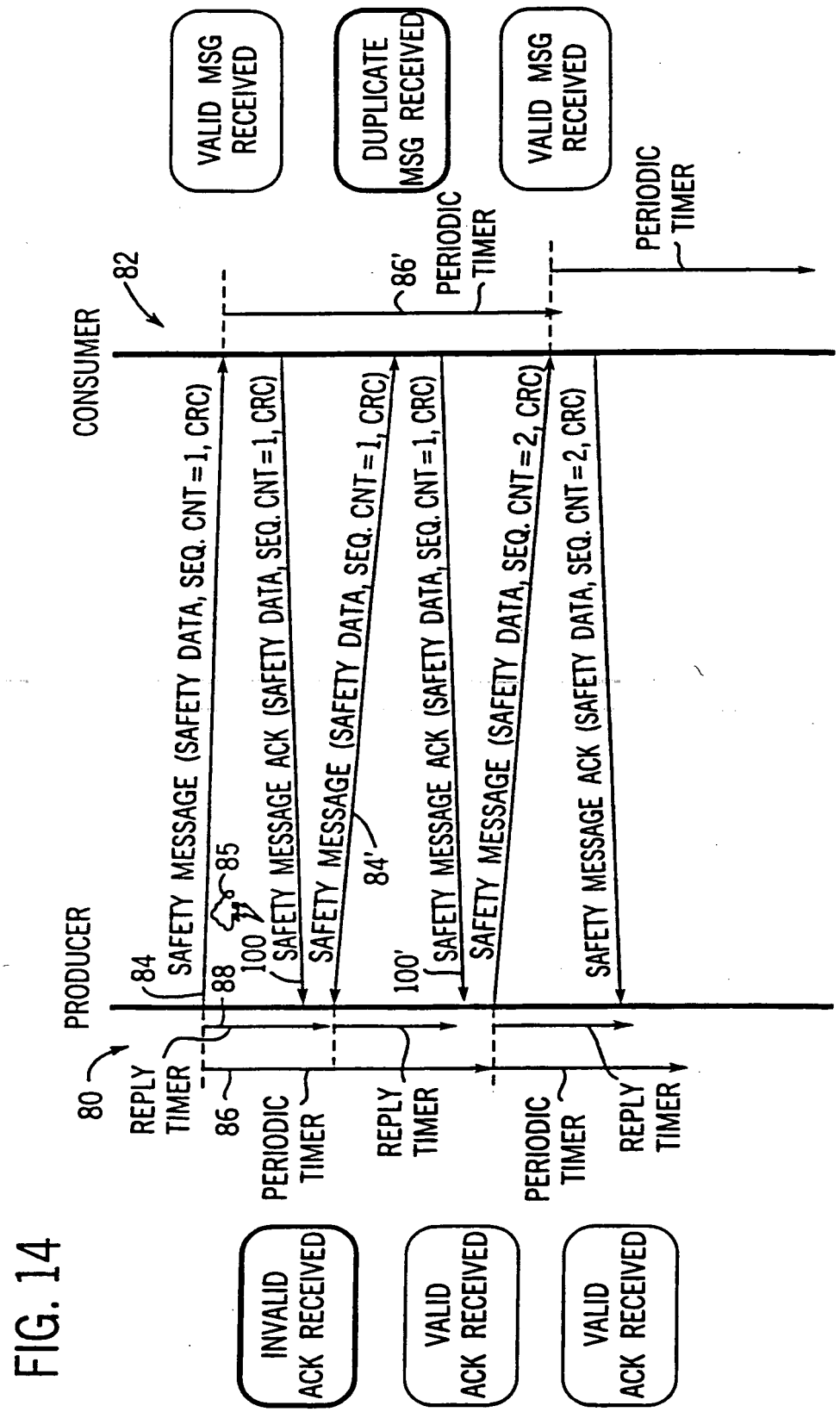

PRODUCER
CONSUMER
80
82
84
84'
85
86
86'
88
100
100'
REPLY TIMER
PERIODIC TIMER
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT = 1, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT = 1, CRC)
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT = 1, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT = 1, CRC)
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT = 2, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT = 2, CRC)
VALID MSG RECEIVED
DUPLICATE MSG RECEIVED
VALID MSG RECEIVED
INVALID ACK RECEIVED
VALID ACK RECEIVED
VALID ACK RECEIVED

FIG. 14

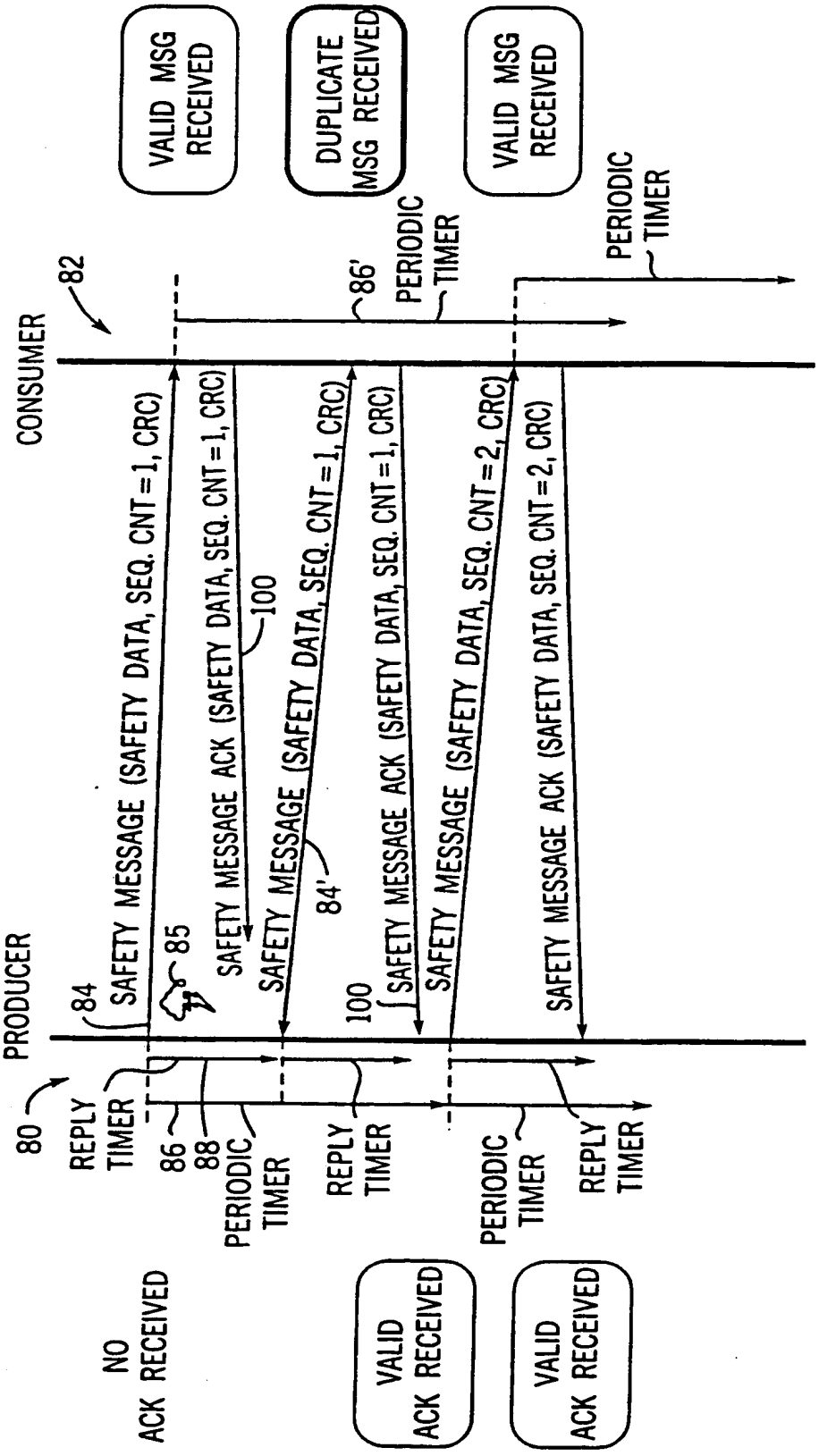

PRODUCER
CONSUMER
82
80
84
85
86
88
86'
84'
100
REPLY TIMER
PERIODIC TIMER
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=1, CRC)
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=1, CRC)
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=2, CRC)
SAFETY MESSAGE ACK (SAFETY DATA, SEQ. CNT=2, CRC)
NO ACK RECEIVED
VALID ACK RECEIVED
VALID ACK RECEIVED
VALID MSG RECEIVED
DUPLICATE MSG RECEIVED
VALID MSG RECEIVED

FIG. 15

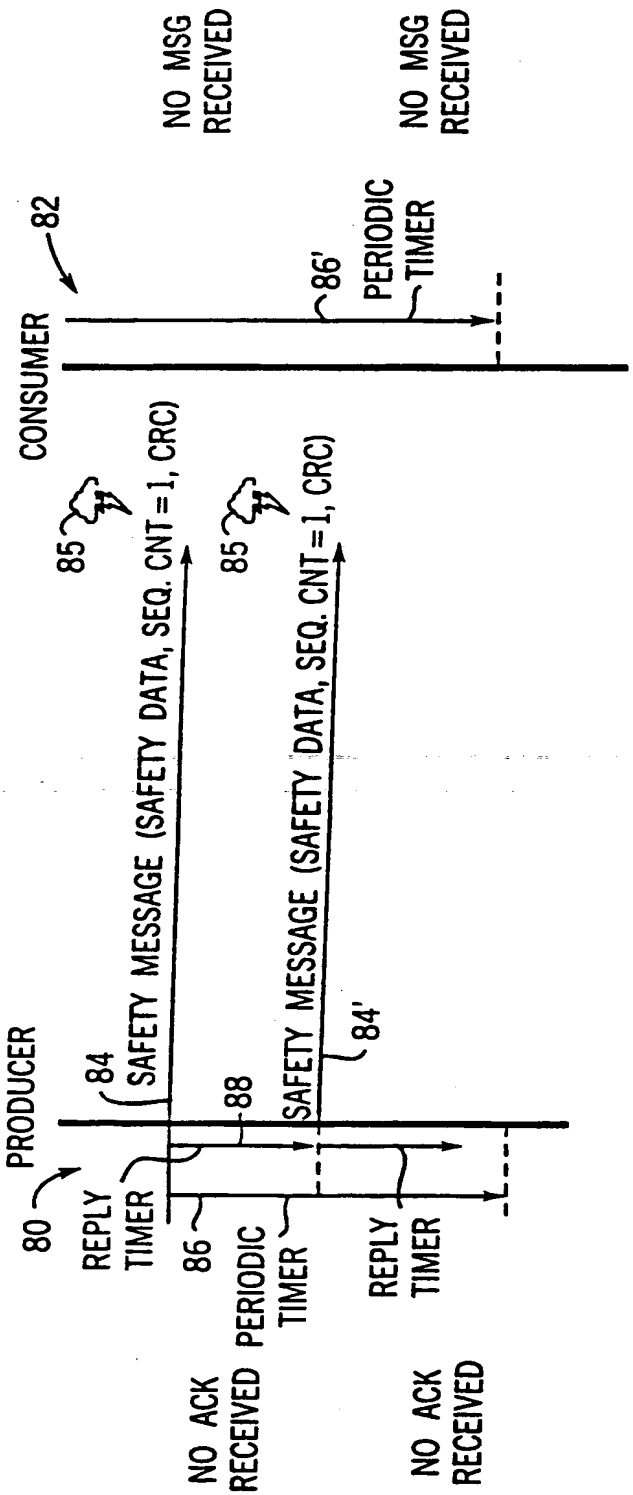

PRODUCER
80
REPLY TIMER
NO ACK RECEIVED
86
PERIODIC TIMER
84
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
88
NO ACK RECEIVED
REPLY TIMER
84'
SAFETY MESSAGE (SAFETY DATA, SEQ. CNT=1, CRC)
85
85
CONSUMER
82
NO MSG RECEIVED
86'
PERIODIC TIMER
NO MSG RECEIVED

FIG. 16

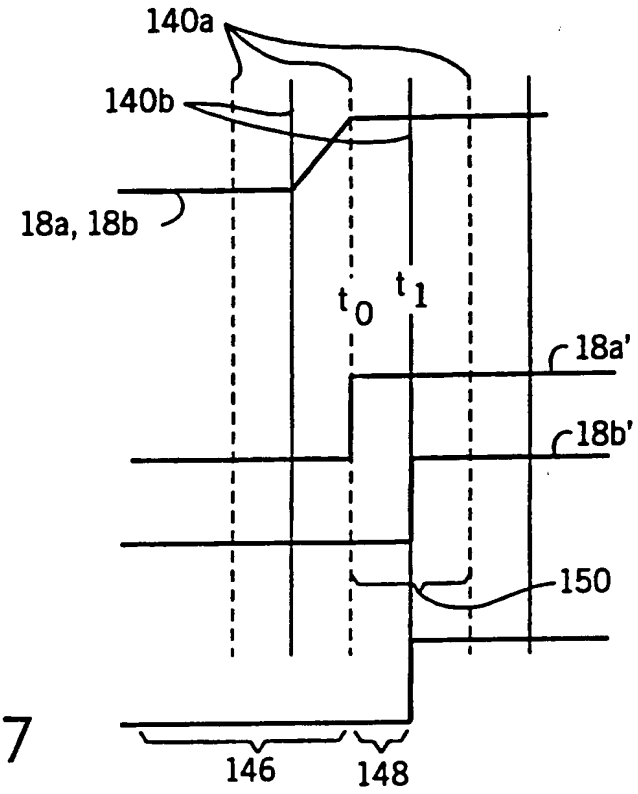

140a
140b
18a, 18b
t_0
t_1
18a'
18b'
150
146
148

FIG. 17

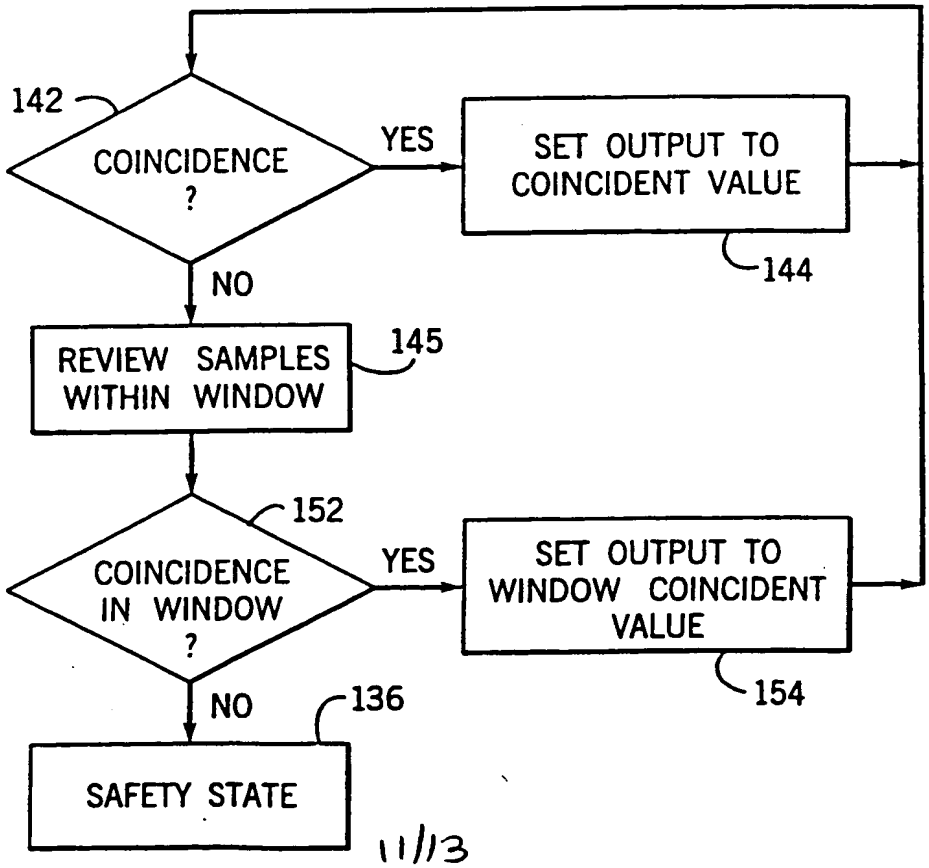

142
COINCIDENCE ?
YES
SET OUTPUT TO COINCIDENT VALUE
144
NO
REVIEW SAMPLES WITHIN WINDOW
145
152
COINCIDENCE IN WINDOW ?
YES
SET OUTPUT TO WINDOW COINCIDENT VALUE
154
NO
136
SAFETY STATE

FIG. 18

FIG. 19

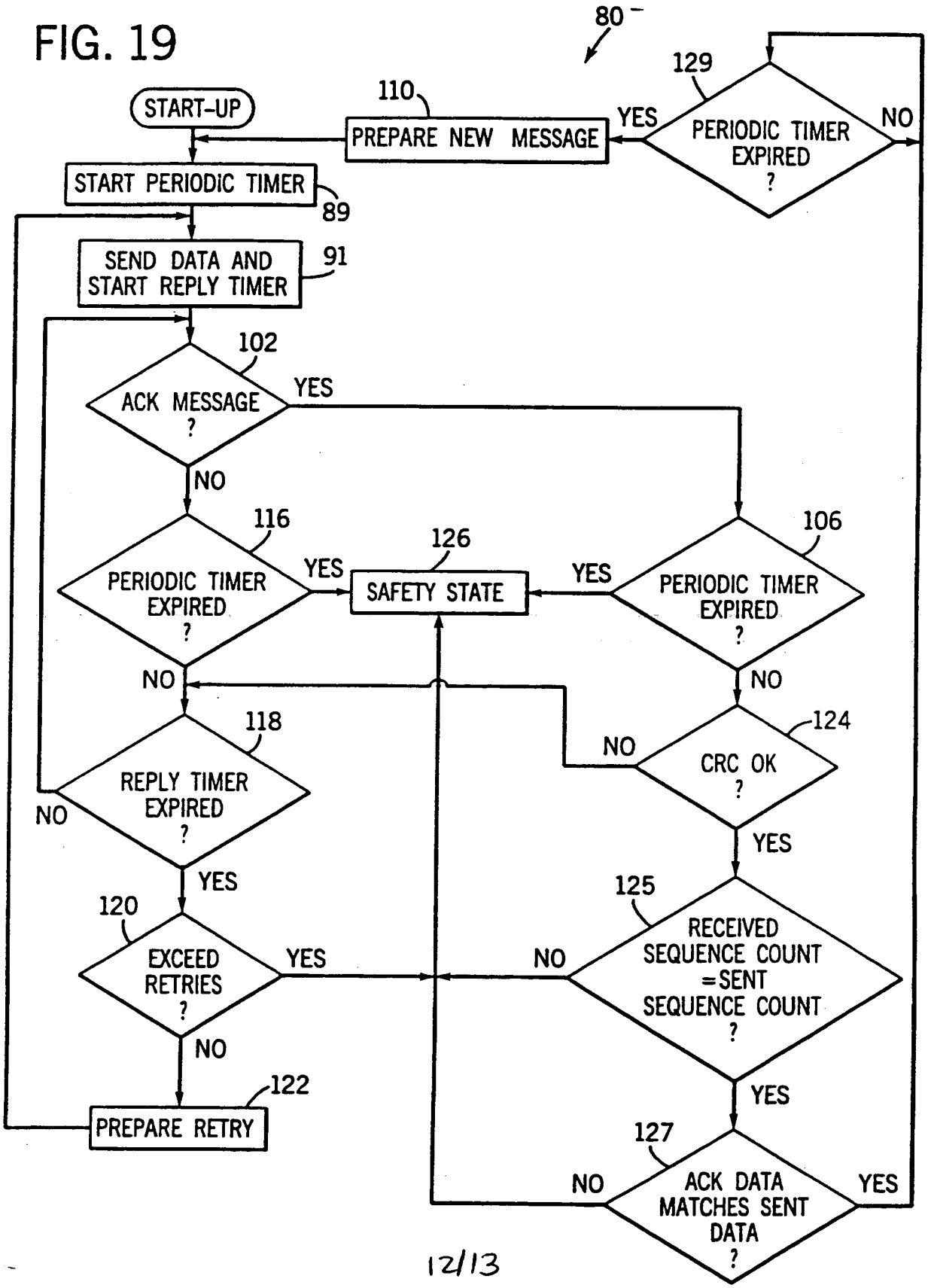
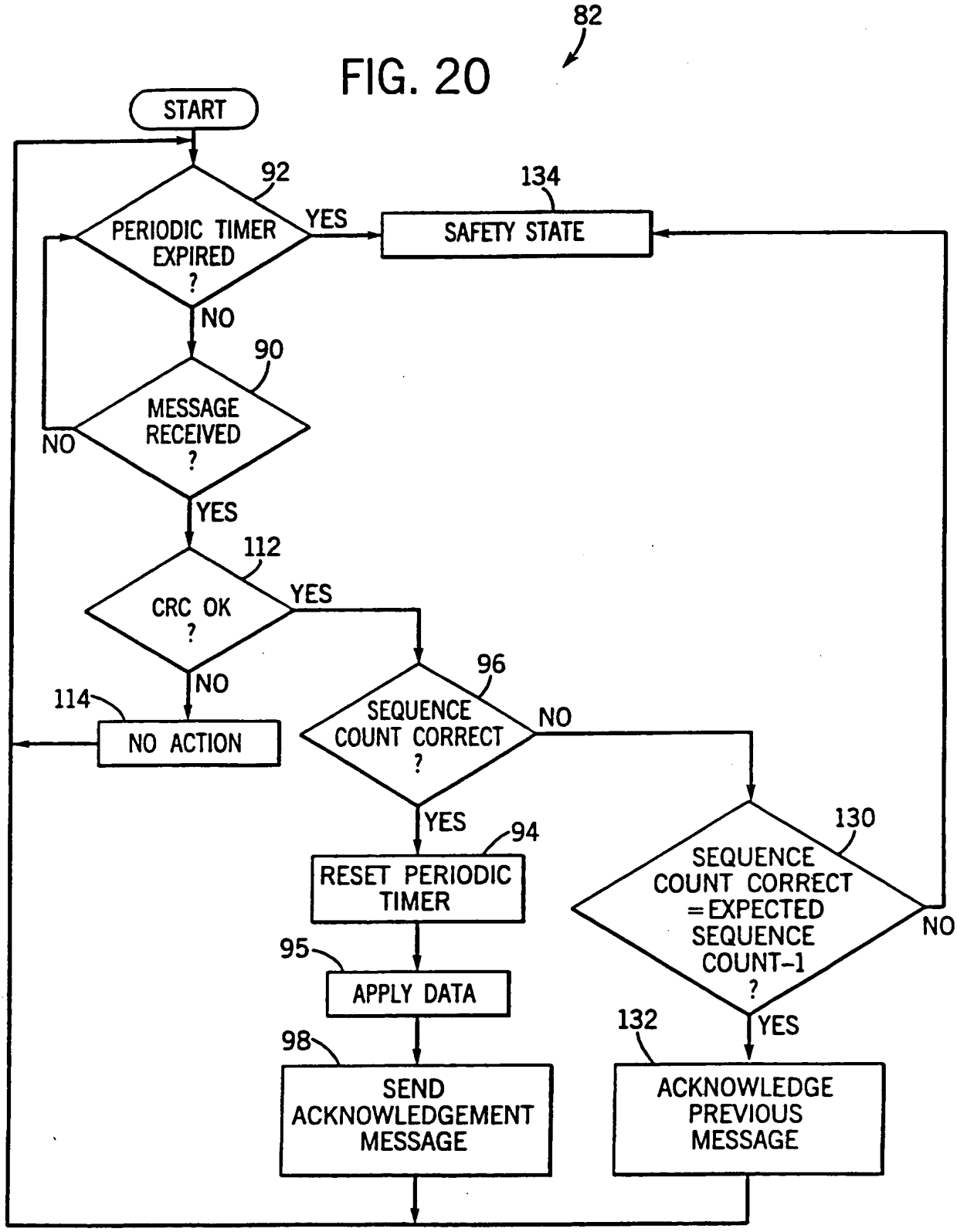

80
START-UP
110
PREPARE NEW MESSAGE
129
PERIODIC TIMER EXPIRED ?
YES
NO
START PERIODIC TIMER
89
SEND DATA AND START REPLY TIMER
91
102
ACK MESSAGE ?
YES
NO
116
PERIODIC TIMER EXPIRED ?
YES
126
SAFETY STATE
106
PERIODIC TIMER EXPIRED ?
NO
NO
118
REPLY TIMER EXPIRED ?
NO
YES
124
CRC OK ?
NO
YES
120
EXCEED RETRIES ?
YES
NO
125
RECEIVED SEQUENCE COUNT =SENT SEQUENCE COUNT ?
NO
YES
122
PREPARE RETRY
127
ACK DATA MATCHES SENT DATA ?
NO
YES

FIG. 20


82
START
92
PERIODIC TIMER EXPIRED ?
YES
134
SAFETY STATE
NO
90
MESSAGE RECEIVED ?
NO
YES
112
CRC OK ?
YES
NO
114
NO ACTION
96
SEQUENCE COUNT CORRECT ?
NO
YES
94
RESET PERIODIC TIMER
95
APPLY DATA
98
SEND ACKNOWLEDGEMENT MESSAGE
130
SEQUENCE COUNT CORRECT = EXPECTED SEQUENCE COUNT-1 ?
NO
YES
132
ACKNOWLEDGE PREVIOUS MESSAGE

INTERNATIONAL SEARCH REPORT

Intern. ial Application No
PCT/US 00/35258

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G05B19/418 G05B19/042

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
PAJ, WPI Data, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	EP 0 977 391 A (LUCENT TECHNOLOGIES INC) 2 February 2000 (2000-02-02) page 3, line 58 -page 10, line 18; figures 1A-7	1,2,11, 26,38
P,A	---	3-10, 12-25, 27-37, 39-49
A	US 5 910 778 A (KLEIN THOMAS L ET AL) 8 June 1999 (1999-06-08) column 5, line 28 -column 10, line 26; figures 1-20	1-49

	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 May 2001	31/05/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Nettesheim, J

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Intern. ial Application No
PCT/US 00/35258

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GLANZER D A ET AL: "Interoperable fieldbus devices: a technical overview" ISA TRANSACTIONS,US,INSTRUMENT SOCIETY OF AMERICA. PITTSBURGH, vol. 35, no. 2, 1996, pages 147-151, XP004020118 ISSN: 0019-0578 page 148, left-hand column, paragraph 2 -page 150, right-hand column, paragraph 2; figures 2-5	1-49

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern. tal Application No
PCT/US 00/35258

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0977391 A	02-02-2000	AU 4113399 A	17-02-2000
		CN 1244080 A	09-02-2000
		JP 2000151535 A	30-05-2000
US 5910778 A	08-06-1999	WO 9711565 A	27-03-1997